محاضرات في مقرر

مبادئ قواعد البيانات

لطلاب (برمجة الحاسوب، علوم الحاسوب، تقنية المعلوماتIT)

بقلم: أ.فهد آل قاسم
مدرس مساعد،قسم علوم الحاسوب
جامعة العلوم والتكنولوجيا ـ فرع إب
fhdalqasem@yahoo.com

المرجع الرئيسي للمادة العلمية هو:

Elmesri & navath, fundamental of database systems

١ مدخل إلى قواعد البيانات

لقد صار مصطلح قاعدة البيانات من أهم مفردات المجتمعات المتقدمة هذه الأيام، وصارت تطبيقات قواعد البيانات جزءا هاما من مكونات أي عملية تجارية أو اقتصادية أو سياسية أو تعليمية، بل مع تطور الويب وانتشار تقنيات الانترنت، صارت قواعد البيانات تمثل حجر الأساس في جميع عمليات تبادل المعلومات، ومكون رئيسي في أي نظام معلومات محوسب، مستفيدة من التطور المناظر في تقنيات الاتصال وشبكات الحاسوب، وقبل أن نتساءل عن ماهية قواعد البيانات، سوف نتأمل عمليا تطبيقاتها الحالية، ونناقش أهم أنواع تطبيقات قواعد البيانات في الواقع.

أنواع تطبيقات قواعد البيانات في العالم الحقيقي:-

■ **التطبيقات التقليدية لقواعد البيانات Traditional Applications:**

وهي تطبيقات مضى على وجودها ما يقارب نصف قرن، ومع ذلك فما زال الاحتياج لها مستمرا، ومازالت تقنياتها تتطور يوميا، ومن أمثلتها قواعد البيانات النصية والرقمية Numeric and Textual Databases، أي قواعد البيانات التي تحتوي على أرقام ونصوص، تخزن فيها، ثم تعالج وتخرج كتقارير حسب التطبيق المستخدم.

■ **التطبيقات الحديثة لقواعد البيانات More Recent Applications:**

وهي التطبيقات التي ظهرت كتطور طبيعي للتطبيقات التقليدية، ونتجت بعد الاحتياج لصيغ أكثر تعقيدا لخزن البيانات، وتقنياتها تعتمد بالطبع على التقنيات التقليدية، ولكنها أكثر تعقيدا وتحتاج إلى حاسبات أكثر تعقيدا، كما أنها تستخدم في المؤسسات والشركات العملاقة، والدول ومؤسساتها الرسمية، وهي تطبيقات كثيرة نذكر منها على سبيل المثال:

١) قواعد بيانات الوسائط المتعددة Multimedia Databases:

وتستخدم لتخزين ومعالجة بيانات في صيغة وسائط متعددة (ملتيميديا)، كالصور والصوت والفيديو، وأبسط مثال عليها موقع اليوتيوب الشهير، الذي يخزن مقاطع الفيديو الخاصة بمستخدميه، ويسمح بتصفحها والإضافة إليها والتعديل وغير ذلك من العمليات اليومية للمستخدمين.

٢) نظم المعلومات الجغرافية Geographic Information Systems (GIS):

تتكون نظم المعلومات الجغرافية من قواعد بيانات معقدة، تشبه نوعا ما قواعد بيانات الوسائط المتعددة، ولكنها تحتوي على تفاصيل جغرافية دقيقة، ومن أبسط أمثلتها البرنامج المعروف باسم (جوجل إيرث).

٣) مخازن أو مستودعات البيانات Data Warehouses:

وهي تطبيقات معقدة تستخدم قواعد البيانات الموجودة، التقليدية أو الحديثة، والتي مضى عليها فترات طويلة، من أجل الحصول على معارف وخبرات لم تكن واردة عند إنشائها للمرة الأولى.

٤) قواعد البيانات النشطة وأنظمة الوقت الحقيقي Real-time and Active Databases:

وهي قواعد بيانات تعمل من أجل تنفيذ عمليات معقدة حساسة للوقت، فالمعالجة ودقتها تكون لها الأولوية على تخزين البيانات وحفظها، بمعنى أن الدقة والوصول تكون أهم خصائصها، وتستخدم في الأنظمة الهندسية والعلمية والطبية المعقدة.

وسيركز المقرر على التطبيقات التقليدية، بإعطاء أساسيات قواعد البيانات النصية والرقمية بالدرجة الأولى، وهذا الموضوع كما أسلفنا هو النواة التي تخدم الدارس في التعامل مع التطبيقات التقليدية من جهة، وفي فهم التطبيقات الحديثة من جهة أخرى.

قاعدة البيانات database:- هي مجموعة مشتركة من البيانات المترابطة والمتجانسة منطقيا، والتي صممت كي تلبي الاحتياج المستمر للمعلومات لمنظمة معينة.

Database is a Shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.

وبالطبع فإن الكلمة (المنظمة) تشير إلى أي جهة تستخدم قواعد البيانات، سواء كان مشروعا تجاريا صغيرا، أو مؤسسة استثمارية متوسطة، أو مجموعة كبيرة من الشركات متعددة الجنسيات، وسواء كانت هذه المؤسسة فردية أو حكومية، وأيا كان مكان قاعدة البيانات في حاسب شخصي صغير، أو في شبكة محلية متوسطة، أو حتى شبكة عالمية كالشبكة العنكبوتية.

البيانات Data :-

هي حقائق (خام) معروفة يمكن أن تسجل ولها معنى ضمني (implicit)، خلافا للمعلومات ذات المعنى الصريح (explicit)، وتحتوي قاعدة البيانات على بياناتها بمعناها الضمني، وبصورة تجعلها قابلة للتحول على معلومات بعد إجراء بعض المعالجات البسيطة أو المعقدة.

إن المعنى الضمني للبيانات، ينتج من كونها بيانات مترابطة، حسب بيئة محددة يطلق عليها العالم الخاص، أو العالم المصغر، **والعالم المصغر mini-world**، هو الجزء من العالم الحقيقي الذي نخزن البيانات عنه في قاعدة البيانات ، مثلاً الجزء المتعلق بدرجات طالب في النظام الجامعي.

حاول سرد مجموعة من الأمثلة عن قواعد بيانات، آخذا بالاعتبار عالمها المصغر الذي تقوم هي بتمثيله، سنعرض تاليا مثال نوضح فيه قاعدة بيانات الجامعة.

نظام إدارة قاعدة البيانات DBMS (Database Management System) :-

هو النظام الذي يسهل إنشاء وصيانة قواعد البيانات المحوسبة، ويأتي عادة على شكل حزمة برمجية، تتكون من برمجيات تقدم للمستخدمين، بمختلف أنواعهم، خدمات الوصول إلى البيانات ومعالجتها، ويكون حلقة وصل بين التطبيق والبيانات، وبين البيانات أيضا ومسئول قاعدة البيانات، أنظر الشكل التوضيحي.

ونظام إدارة قاعدة البيانات هو أحد مكونات نظام قاعدة البيانات DBS، والموضح في الشكل، ويتكون نظام قاعدة البيانات من:

١. **Users / programmers المستخدمين والمبرمجين:**

وهو المستخدم النهائي، ويدخل في ذلك المبرمج programmer ومدير قاعدة البيانات DBA، رغم اختلاف المهام وطرق الوصول، وسوف نتعلم أنواع المستخدمين ووظائفهم تاليا.

٢. **Application programs / Queries التطبيقات البرمجية/ الاستعلامات:**

يتم استرجاع البيانات من قاعدة البيانات إما مباشرة باستخدام لغة الاستعلامات، أو عن طريق واجهات التطبيق البرمجي، وهي واجهات إدخال وواجهات إخراج البيانات والمعلومات، والمطورة بواسطة إحدى لغات برمجة قواعد البيانات المعروفة.

٣. **نظام إدارة قاعدة البيانات DBMS**: من أمثلته (ORACLE، SQLSERVER، MYSQL....etc).

٤. **قاعدة البيانات** نفسها: والتي تتكون من تعريفات البيانات meta-data، ومن البيانات الخام المخزنة ، كل على حده.

نموذج يوضح نظام قاعدة البيانات Database System Model :-

لاحظ، في الشكل التالي، موقع DBMS الذي يتوسط تطبيقات المستخدم وقاعدة البيانات، وهذا الموقع يوضح الأهمية القصوى له، حيث يربط بين البيانات الخام وبين تعريفاتها من جهة البيانات، ويساعد كذلك في إنشاء قواعد البيانات ومعالجتها، ومن جهة أخرى يقوم بالتعامل مع تطبيقات المستخدم، واستقبال المدخلات وتخزينها، أو

تزويده بالنتائج والمخرجات التي يطلبها.

وبطبيعة الحال فإن المبرمج ومدير قاعدة البيانات، هما من يستخدم الـ DBMS (مدير قاعدة البيانات)، ويعده لتنفيذ تلك المهام التي تتكامل مع دور التطبيق البرمجي.

أما قاعدة البيانات نفسها، والتي يديرها الـDBMS، فتتكون من:

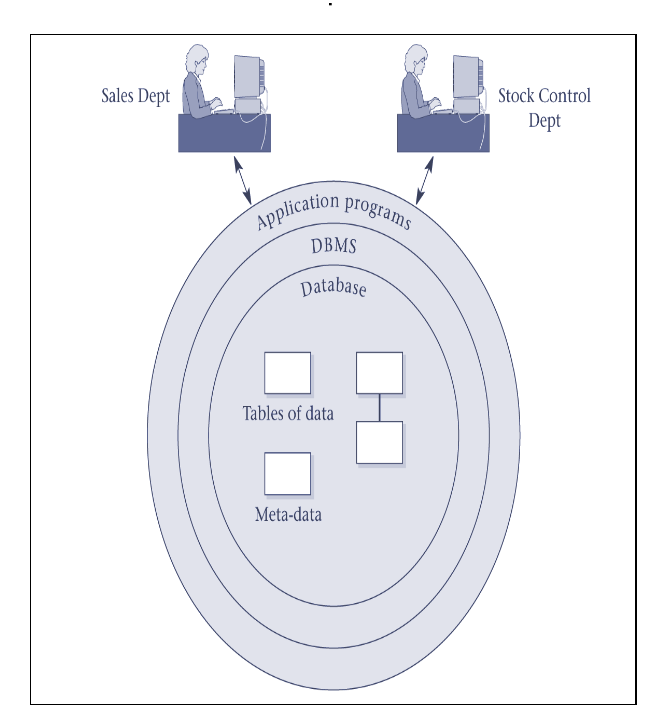
١) Data definitions تعريفات البيانات:

وهي البيانات المخزنة عن البيانات، أو هي معلومات عن بنية البيانات وليس البيانات نفسها، وتسمى أيضا بـ meta-data.

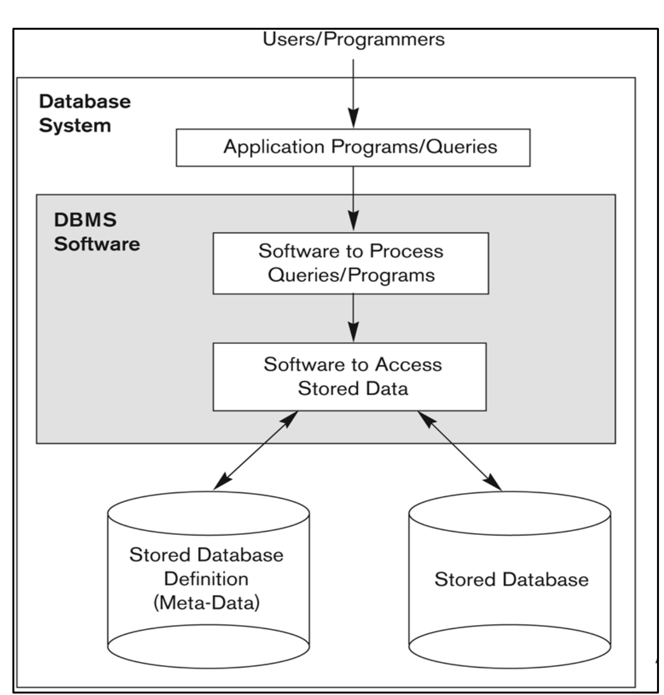
٢) Stored database قواعد البيانات المخزنة:

وهي البيانات نفسها، التي تفصل عن تعريفاتها، من أجل مرونة إجراء اي تعديل او تحديث للبيانات، مما يقلل من زمن المعالجة ومن مساحة التخزين، أكثر مما يمكن.

- ب-



-أ -



شكل يوضح طريقتين لعرض نظام قاعدة البيانات (المستخدم، التطبيق البرمجي، الـDBMS ، البيانات)

<u>مثل حول قاعدة البيانات :</u>

العالم المصغر للمثال هو جزء من بيئة جامعة، وعند تصميم النموذج الأولي للبيانات CDM (conceptual data model)، تظهر لنا مجموعة من المكونات، حيث يسمى كل مكون بالكينونة entity، وبتجميع هذه الكينونات وخصائصها نحصل على قاعدة البيانات المطلوبة، ولدينا هنا **بعض** كينونات <u>**العالم المصغر**</u>:**طالب، مقرر دراسي**، تمارين مقرر، **قسم علمي، مدرس** .. إلخ.

<u>Mini-world for the example:</u>Part of a UNIVERSITY environment.

<u>Some mini-world *entities*:</u>

STUDENTs	**طالب**
COURSEs	**مقرر**
SECTIONs (of COURSEs)	تمارين
(academic) DEPARTMENTs	**قسم علمي**
INSTRUCTORs	**مدرس**

يتكون **النموذج الأولي** للبيانات، بالإضافة إلى الكينونات وصفاتها، من مجموعة من العلاقات، والتي تمثل علاقاتها الحقيقية في الواقع، وسنتطرق لهذا النموذج بالتفصيل لاحقا.

بعض العلاقات relationships في العالم المصغر :-

– المقرر العملي يتبع مقرر دراسي .

– المدرس يدرس مقرر دراسي .

– الطالب يأخذ مقرر عملي .

– المقرر الدراسي يقدم بواسطة قسم علمي .

وتكون حصيلة النموذج الأولي للبيانات، بعد تطبيق مجموعة معينة من الإجراءات، قاعدة بيانات مادية تتكون من جداول مترابطة فيما بينها وفق النموذج العلائقي، ولذا يحتوي الشكل التالي على أربعة جداول هي : (مقرر ، مقرر عملي ، درجات ، مقررسابق).

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

مجموعة من الجداول تمثل قاعدة بيانات بسيطة لجزء من نظام جامعة

كتلوج نظام إدارة البيانات DBMS Catalog :-

هو جزء من الـ DBMS الذي **يحتفظ بمعلومات عن الجدول**، وليس بيانات الجدول نفسه ، **أي عن** بنية الجدول، **مثل اسم الجدول**، أنواع البيانات فيه، أي أنه يخزن تعريفات البيانات (data definitions) الموجودة في قاعدة البيانات والتي أسميناها meta-data.

ويطلق على هذا الكتالوج أكثر من تسمية، منها بنية قاعدة البيانات ومنها تعريفات قاعدة البيانات وكذلك مخطط السكيما لقاعدة البيانات كما سيأتي لاحقا، وبغض النظر عن التسمية، فإن على مصمم قاعدة البيانات الاهتمام بعملية التعريف، التي ينتج عنها هذا الكتالوج، فهي حجر الأساس في تطوير قواعد البيانات الحديثة، مقارنة بالنظم القديمة التي لم تكن تستخدم هذه التقنية.

مستخدمو قاعدة البيانات Data Users :-

هم الأشخاص الذين يستخدمون أو يديرون قاعدة البيانات ومحتوياتها ، والذين يصممون ويطورون ويحافظون على تطبيقات قواعد البيانات ، وكذلك بالنسبة لل DBMS .

ويصنف المستخدمون إلى قسمين كل قسم يتكون من مجموعة من التصنيفات هي:

١-المستخدم غير المباشر ويسمون مجازا workers behind the scene (أي عاملون خلف المنصة): ومن أمثلتهم مدير قاعدة البيانات، ومصمم قاعدة البيانات.

مديرو قاعدة البيانات database administrators :

مسئوليتهم في منح حقوق وشرعية الوصول إلى قاعدة البيانات ، وفي تنسيق ومراقبة الاستخدام ، والحصول على موارد البرمجيات والعتاد ، والتحكم في الاستخدام ومراقبة كفاءة العمليات .

مصممو قاعدة البيانات database designers :

مسئوليتهم في تعريف المحتوى وكتالوج قاعدة البيانات وكذلك شروطها ووظائفها ومعاملاتها, ويجب أن يتواصل مصممو قاعدة البيانات مع المستخدم النهائي end-user ويجب عليهم فهم احتياجاته، وعكسها في قاعدة البيانات المصممة.

٢-المستخدم المباشر ويسمون مجازاً a actors on the scenes (ممثلون على المنصة): وهو المستخدم النهائي.

المستخدمون النهائيون end –users :

يستخدم هذا النوع من المستخدمين البيانات للاستعلام وإخراج التقارير وبعضهم يحدث محتوى قاعدة البيانات,ويصنف هذا النوع إلى المستخدمين التاليين : (١- المستخدم العادي ٢- المستخدم النادر ٣- المستخدم المركب ٤- المستخدم الوحيد).

١-المستخدم النادر casual user : يصل المستخدم إلى قاعدة البيانات نادراً عند الحاجة .

٢- المستخدم العادي naive user : (أو المستخدم الدائم parametric user)

وهذا هو النوع الذي يقوم بالقسم الأكبر من أعمال المستخدم النهائي حيث يقوم بالإجراءات الثابتة المعرفة مسبقاً ، ويقوم بتنفيذها على قاعدة البيانات. ومن أمثلة ذلك (الموظف الذي يقوم بالحجز في نظام حجوزات الطيران ويجلس على النظام فترة طويلة ويكون أحياناً هناك موظف آخر يبادله مناوبات على نفس العمل).

٣- المستخدم المركب sophisticated user : وهو المستخدم ذو المهام المعقدة كمحللي النظام ز العلماء المتخصصين والمهندسين ويكون ليه إطلاع شامل وتآلف مع قدرات النظام ، ويستخدمون عادةً برمجيات وأدوات جاهزة لإنجاز أعمالهم .

٤- المستخدم الوحيد stand-alone user : غالباً يقوم هذا المستخدم بصيانة قاعدة البيانات الشخصية بواسطة برمجيات جاهزة ، مثال ذلك (مستخدم برنامج ظرائب بسيط حيث يقوم ببناء واستخدام قاعدة بيانات خاصة به ، وكذلك بالنسبة لنظام دليل الهاتف .

بعض مميزات استخدام طرق قواعد البيانات Advantages of Using the Database Approach :

-التحكم بتكرار البيانات redundancy عند تخزينها ، وضبط الوصول غير الشرعي (غير المصرح) للبيانات ، وتقديم بنية تخزين مفهرسة لتسريع المعالجة والاستعلام .

- تقديم خدمات النسخ الاحتياطي والاستعادة backup، recovery .

- تقديم واجهات متعددة لمختلف المستخدمين ، تمثيل علاقات معقدة بين البيانات .

- الدفع نحو تطبيق شروط سلامة البيانات وغير ذلك من المميزات منها (مرونة تغيير بنية البيانات ، فرصة التوجه نحو المعيارية standard ....الخ).

التطور التاريخي لقواعد البيانات :

١) تطبيقات قواعد البيانات القديمة : النموذج الشبكي ، النموذج الهرمي التي كانت موجودة في الستينات.
٢) نظم قواعد البيانات العلائقية : بدأت في السبعينات وقد طبقت في الثمانينات عبر بعض الـ DBMS التجارية.
٣) التطبيقات الحديثة كائنية التوجة Object-Oriented DB .
٤) البيانات في التجارة الإلكترونية وتطبيقات الويب : ، languages scripts لغة HTML وXML

ورغم الطورات المتسارعة في نظم قواعد بيانات، إلا أن النظام ألعلائقي مازال هو النظام المعمول به في حزم الـ BDMS المختلفة، وحتى النظم الأحدث فقد طورت عنه.

متى لا ينصح باستخدام DBMS؟

-إذا كانت تكلفة برمجيات الـ DBMS أغلى من جدوى النظام نفسه .

- عندما لا تكون برمجيات الـ DBMS ضرورية كأن تكون قاعدة البيانات واضحة وبسيطة وثابتة .

- عندما تكون قاعدة البيانات أعقد من الـ DBMS المتاح .

٢. نماذج البيانات
Data Models

نموذج البيانات Data Model:

هو مجموعة من المفاهيم التي تقوم بوصف بنية قاعدة البيانات والشروط أو القيود التي تسير قاعدة البيانات وفقها.

عمليات نموذج البيانات Data Model Operations:

هي العمليات التي تحدد آلية استرجاع البيانات وتحديثها من قاعدة البيانات، وذلك بالرجوع إلى مفهوم النموذج التي تسير وفقه، وتشمل هذه العمليات نوعين هما العمليات الاساسية basic operations التي تعرف من ضمن النموذج، والعمليات التي يعرفها المستخدم user-defined operations.

بنية البيانات وشروطها هي بعض التفاصيل المتعلقة بطبيعة تلك البيانات، فمثلا جدول بيانات الموظف في قاعدة بيانات موظفين توصف بنيته structure بأنه يتكون من الاسم وتاريخ الميلاد وغيرها من الحقول، أما شروطه أو القيود constraints فقد تكون مثلا أن لا يزيد الاسم عن ٥٠ حرفا، وان لا يقل تاريخ الميلاد عن ١٩٨٠م، على سبيل المثال، وغيرها من القيود التي تطبق على بنية قاعدة البيانات.

<u>تصنيف نماذج البيانات Categories of data models:</u>

تقسم نماذج البيانات إلى ثلاثة تصنيفات رئيسية هي:

١. نماذج البيانات الأولية (أو المفاهيمية) Conceptual data models:
تقدم هذه النماذج مفاهيما قريبة من طريقة المستخدم في استقبال البيانات، ولذلك يسمى نموذج البيانات من هذا النوع بنموذج المستوى الأعلى، أي المستوى الذي يفهمه المستخدم، في هذا النموذج يتم تقديم توصيف لقاعدة البيانات بحسب رؤية المستخدم وشروطه وعملياته.

٢. نماذج البيانات المادية (أو الفيزيائية) Physical data models:
تقدم هذه النماذج مفاهيما تصف تفاصيل تخزين البيانات في الحاسوب بصورتها المادية، ولذلك فهي تسمى أيضا بنماذج البيانات منخفضة المستوى، أي المستوى الذي يفهمه الحاسوب، ويحتوي على توصيفات قاعدة البيانات القريبة من لغة الآلة.

٣. نماذج البيانات التنفيذية (أو التمثيلية) Implementation (representational) data models:
تقدم هذه النماذج المفاهيم المتوسطة، والتي تقع بين النموذجين أعلاه، وذلك بعمل توازن بين رؤية المستخدم التي في النموذج الأولي وبين تفاصيل تخزين البيانات في الحاسوب (النموذج المادي)، وهذه هي مهمة مصمم قاعدة البيانات في تحويل النموذج الأولي إلى نموذج مادي.

<u>مخطط البيانات وأمثلة البيانات Schemas versus Instances:</u>

مخطط قاعدة البيانات Database Schema:

هو مخطط يوصف قاعدة البيانات، حيث يشمل وصف لبنية البيانات فيها، وشروط تلك البيانات أو القيود التي يجب أن تطبق عليها، ويمكن اعتباره كتالوجا أو قاموسا يصف تلك البيانات، بغض النظر عن البيانات نفسها.

المخطط الرسومي Schema Diagram :

هو مخطط قاعدة البيانات ولكن بشكل رسومي لعرض بعض مميزات وخصائص المخطط.

بنية المخطط Schema Construct:

هي المكونات التي يتكون منها مخطط قاعدة البيانات، أو هي الكائنات التي تكون المخطط مثلا في قاعدة بيانات جامعة لدينا بعض مكونات مخطط قاعدة البيانات مثل **الطالب ، المقرر** ..وغيره

مثيل (أو تفاصيل) قاعدة البيانات Database Instance:

هي البيانات الفعلية المخزنة في قاعدة البيانات في لحظة معينة من الزمن، وتسمى كذلك حالة أو امتداد قاعدة البيانات (db state)، ويسمى مثيل لأنه يحتوي على تفاصيل لنفس الكيان **المحدد في** قاعدة البيانات، فمثيل الكيان طالب هي بيانات طالب معين..وهكذا، وتترجم أيضا إلى وحدة الكيان instance entity كما سيأتي.

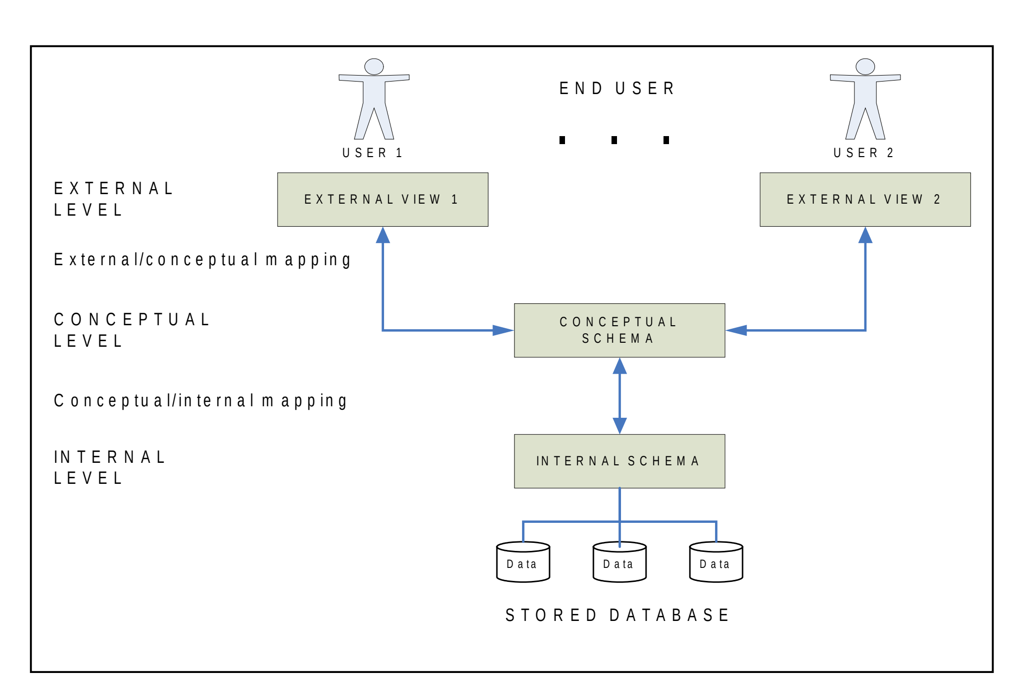
المعمارية الثلاثية لمخطط قاعدة البيانات (معمارية الإسكيما) Three-Schema Architecture:

يتكون مخطط قاعدة البيانات من ثلاثة مستويات، كل مستوى يستخدم واحدا او إثنين من نماذج البيانات المذكورة سابقا، وهذه المستويات موضحة بالشكل أدناه، وهي:

١. **المخطط الداخلي Internal schema**: يوجد هذا المخطط في المستوى الداخلي لقاعدة البيانات، ويقوم هذا المخطط بوصف البنية المادية أو الفيزيائية لقاعدة البيانات ومسار الوصول إليها، ويستخدم هذا المخطط نموذج البيانات الفيزيائي.

٢. **المخطط الأولي (أو المفاهيمي) Conceptual schema**: يقوم هذا المخطط بوصف بنية التعريفات والقيود لقاعدة البيانات بأكملها لمجتمع من المستخدمين، باستخدام نموذج البيانات التمثيلي أو نموذج البيانات الأولي.

٣. **المخطط الخارجي External schemas**: يصف هذا المخطط مختلف المشاهد التي بواسطتها يرى المستخدم قاعدة البيانات، وعادة يستخدم هذا المخطط نفس نموذج البيانات الذي يستخدمه المخطط الأولي.



شكل يوضح المعمارية الثلاثية للاسكيما

إستقلالية البيانات Data Independence:

تتحقق استقلالية البيانات، بالقيام بفصل البيانات التي يراها المستخدم عن بنيتها الداخلية، ويتم ذلك على مرحلتين هما:
الاستقلالية المنطقية للبيانات Logical Data Independence:
هي القدرة على تغيير المخطط الأولي لقاعدة البيانات conceptual schema، بدون تغيير المخططات الخارجية لها ولا تغيير التطبيقات البرمجية عليها.
الاستقلالية الفيزيائية للبيانات Physical Data Independence:
هي القدرةعلى تغيير المخطط الداخلي لقاعدة البيانات internal schema، دون إجراء أي تغيير على المخطط الأولي (المفاهيمي).
يتم دعم إستقلالية البيانات بشقيها المادي والمنطقي بواسطة برمجيات الـ(DBMS)، أو مايسمى بنظام إدارة قاعدة البيانات database management system الذي يمكن أن يكون مركزيا أو موزعا.

مدير قاعدة البيانات المركزي Centralized DBMS:

يقوم الـ (DBMS) في هذه الحالة بدمج كل مكوناته في نظام واحد، العتاد والبرمجيات والتطبيقات، وكذلك برمجيات واجهات المستخدم.

مدير قاعدة البيانات الموزع Distributed DBMS:

ويسمى في هذه الحالة بنظام قواعد بيانات المزود و العميل Client Server DBMS ، لأنه لا يوزع جميع مكونات نظام قواعد البيانات ولكنه يوزع فقط مجموعة من مخدمات (مزودات) قواعد البيانات التي تدعم مجموعة من العملاء أو الطرفيات Terminals.

فئات أو أصناف الـ DBMSs:

توجد في سوق العمل مجموعة كبيرة من حزم الـDBMS ، ولكنها تختلف من جهة نموذج البيانات المستخدم، وكفاءة التقنيات المطبقة، وتكلفة الشراء وغيرها، كما يلي:

١. **حسب نموذج البيانات المستخدم Based on the data model used**
وتصنف حزم الـDBMS إلى نماذج البيانات التالية:

التقليدية: مثل النموذج العلائقي كما سيأتي، أو الاقدم منه كالنموذج الشبكي او الهيكلي.

الحديثة: مثل النموذج الكائني، أو الكائني العلائقي.

نموذج لغة XML العلائقي الموسع.

٢. **حسب عدد المستخدمين Number of users**: من حزم الـDBMS ما يدعم المستخدم الوحيد مثل نظام الأكسس من مايكروسوفت، أو متعددة المستخدمين على جهاز وحيد مثل بعض أنظمة إدارة قاعدة البيانات كالأوراكل والـ(ماي سكيول) وغيرها.
٣. **عدد المواقع Number of Sites**: قد يتعدد المستخدمون على جهاز وحيد في هذه الحالة يكون النظام مركزيا كما سبق، جهاز حاسوب واحد مع قاعدة بيانات واحدة، وقد يكون تعدد المستخدمين على أكثر من جهاز حاسوب، وفي هذه الحالة يكون النظام موزعا (أكثر من حاسوب أكثر من قاعدة بيانات).

٣. منهجية النموذج العلائقي Relational Model Approach
نموذج الكينونة/العلاقة (ERD):

هو نموذج **عالي المستوى** يقوم بعرض بناء البيانات (high-level conceptual data model)، ويتم استخدام هذا النموذج أثناء مرحلة التصميم المفاهيمي للنموذج الأولي (Conceptual Model)، **وينتج عن ذلك النموذج الأولي لقاعدة البيانات**، والذي عن طريقة نقوم بتصميم **مخطط قاعدة البيانات database schema**، ويتم تمثيل بناء البيانات والقيود المطلوبة عليها باستخدام أشكال رسومية سهلة ومحددة.

مكونات نموذج الكينونة/العلاقة (Entity-Relationship Diagram)

الكيان أو الكينونة (Entity):

هو الوحدة الأساسية التي يتم تمثيلها بنموذج الكينونة/العلاقة (ER) ويشير هذا الكيان إلي "شيء" حقيقي في الحياة سواء كان له وجود فعلي مثل (طالب – موظف – سيارة - ... الخ) أو وجود مظفي مثل (شركة – وظيفة – مقرر - ... الخ).

وحدة الكيان (Entity Instance):

يمكن أنه يوجد لكل كيان مجموعة غير محددة من الكيانات الشبيهة، ، يسمى كل واحد منها بالوحدة أو المثيل **Instance**، وهي وحده تحمل نفس صفات الكيان وتختلف عن الكيانات الأخرى بالمفتاح كما سيأتي، فلو نظرنا إلى الكيان على أساس انه جدول، وان صفاته هي أعمدة هذا الجدول، فإن كل صف من هذا الجدول، يعتبر مثيلا للجدول، وتتشابه الوحدات أو المثائل في كونها تحمل نفس الصفات ولكن ببيانات مختلفة.

مفتاح الكيان entity key: لكل كيان مفتاح عبارة عن صفة أو صفات تعرف الكيان بطريقة وحيدة.

ويتم تمثيل الكيان باستخدام شكل مستطيل يكتب داخله اسم الكيان أو الكينونة، كما هو مبين في شكل ١٤.

شكل يمثل الكيان بمستطيل في مخطط ERD

الصفة (Attribute):

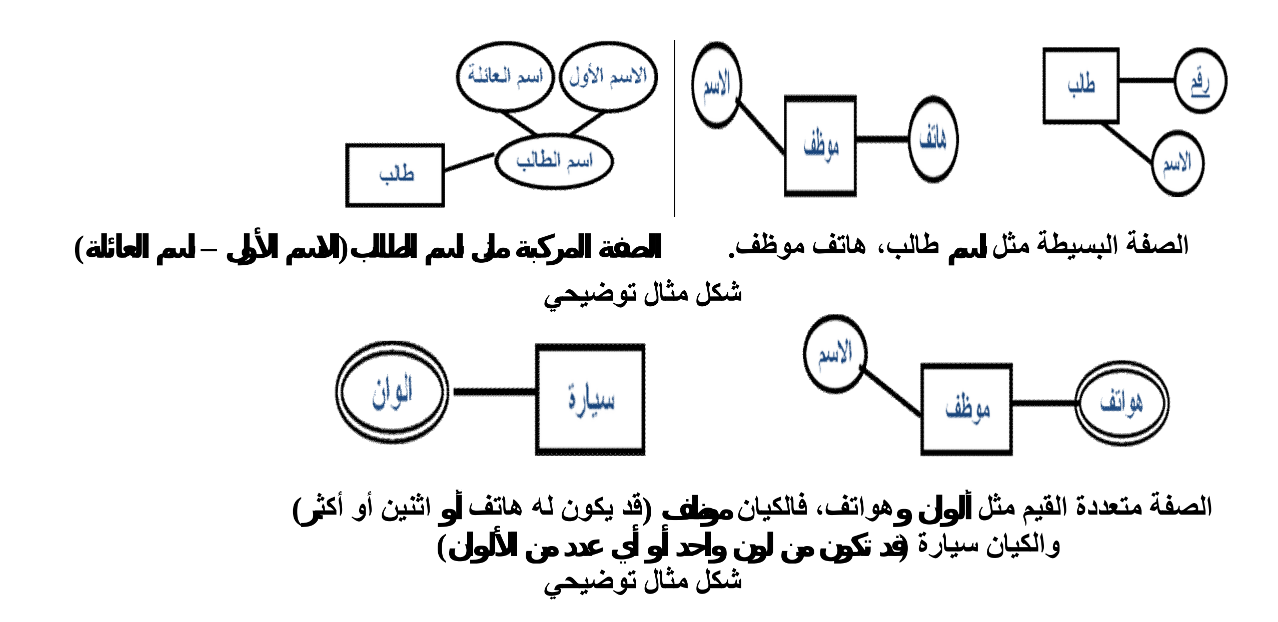
هي صفة معينة تصف الكيان وتكون تابعة له في مخطط ERD، مثل اسم الموظف ، عمر الطالب، مرتب موظف، درجة طالب، عدد الساعات الدراسية لمقرر، لاحظ أن كل صفة تتبع كينونة أو كيانا معينا.

الصفة يمكن أن تكون صفة بسيطة – أو صفة مركبة، الصفة يمكن أن تكون أحادية القيمة – أو صفة متعددة القيم، ويوضع خط تحت اسم الصفة أو الصفات التي تمثل مفتاح الكيان، وكذلك لدينا الصفة المشتقة derived attribute، وهي صفة بسيطة يمكن الحصول عليها من صفة بسيطة أخرى، مثل الصفة عمر الطالب التي يمكن الحصول عليها من تأريخ الميلاد، أو الصفة معدل الطالب التي يمكن الحصول عليها بالطرق الحسابية المعروفة، يتم تمثيل الصفة باستخدام شكل بيضاوي، والصفة متعددة القيم يتم تمثيلها بالشكل البيضاوي المزدوج كما هو موضح بالشكل ١٥.

الصفة المشتقة derived attribute	صفة مركبة Composite Attribute	صفة متعددة القيم Multivalve Attribute	صفة بسيط Simple Attribute

شكل تمثيل الصفات المختلفة للكيان

والأشكال (١٦و١٧) توضح أمثلة للصفات (Attributes) مرتبطة مع كياناتها.



الصفة البسيطة مثل اسم طالب، هاتف موظف.   الصفة المركبة مثل اسم الطالب (الاسم الأول – اسم العائلة)

شكل مثال توضيحي

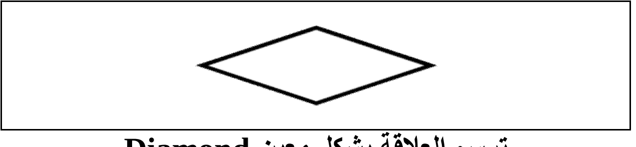
الصفة متعددة القيم مثل ألوان وهواتف، فالكيان موظف (قد يكون له هاتف أو اثنين أو أكثر)
والكيان سيارة (قد تكون من لون واحد أو أي عدد من الألوان)

شكل مثال توضيحي

العلاقات Relationships:

العلاقة (R) بين مجموعة من الكيانات (E1,E2, ...En) هي مجموعة تمثل الارتباطات بين هذه الكيانات، كل وحدة (Instance) في العلاقة (R) هي عبارة عن اتحاد أو ارتباط بين الكيانات المرتبطة بهذه العلاقة, بحيث أن هذه الوحدة (Instance) تمثل بصف واحد من كل كيان مشارك في العلاقة.

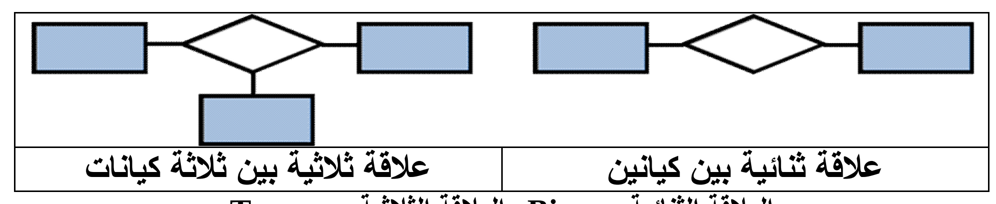
في نموذج الكينونة/العلاقة، يجب أن يتم تمثيل المرجعية من كيان إلى كيان آخر، باستخدام "علاقة" وليس كصفة في الكيان، ويتم تمثيل العلاقة في نموذج الكينونة/العلاقة باستخدام شكل المعين (Diamond shape)، والذي يرتبط مع الكيانات بخطوط مستقيمة وهذا يتم تمثيله بالشكل١٨، ويكتب داخل المعين اسم العلاقة بصيغة فعل (غالبا)، ويمكن أن تتحول العلاقة فيما بعد إلى حقل في جدول، ويمكن أن تصاغ كجدول منفصل، وذلك حسب نوع العلاقة ودرجتها كما سيأتي.



ترسم العلاقة بشكل معين Diamond

درجة العلاقة (Degree of Relationship):

لكل علاقة درجة، وتتحدد هذه الدرجة بعدد الكيانات المرتبطة بهذه العلاقة (ثنائية – ثلاثية - ....)، وسنهتم فيما يلي أكثر بالعلاقات الثنائية.



العلاقة الثنائية Binary والعلاقة الثلاثية Ternary

نوع العلاقة (Cardinality Ratio) :

المصطلح Cardinality Ratio يعني نسبة الارتباط بين وحدات الكيان، التي ترتبط بنفس العلاقة، وفي العلاقة الثنائية بين كيانين، نوع العلاقة هو عدد الوحدات (Instances) في العلاقة التي يمكن أن يشترك فيها الكيان وهي ثلاثة الأنواع:

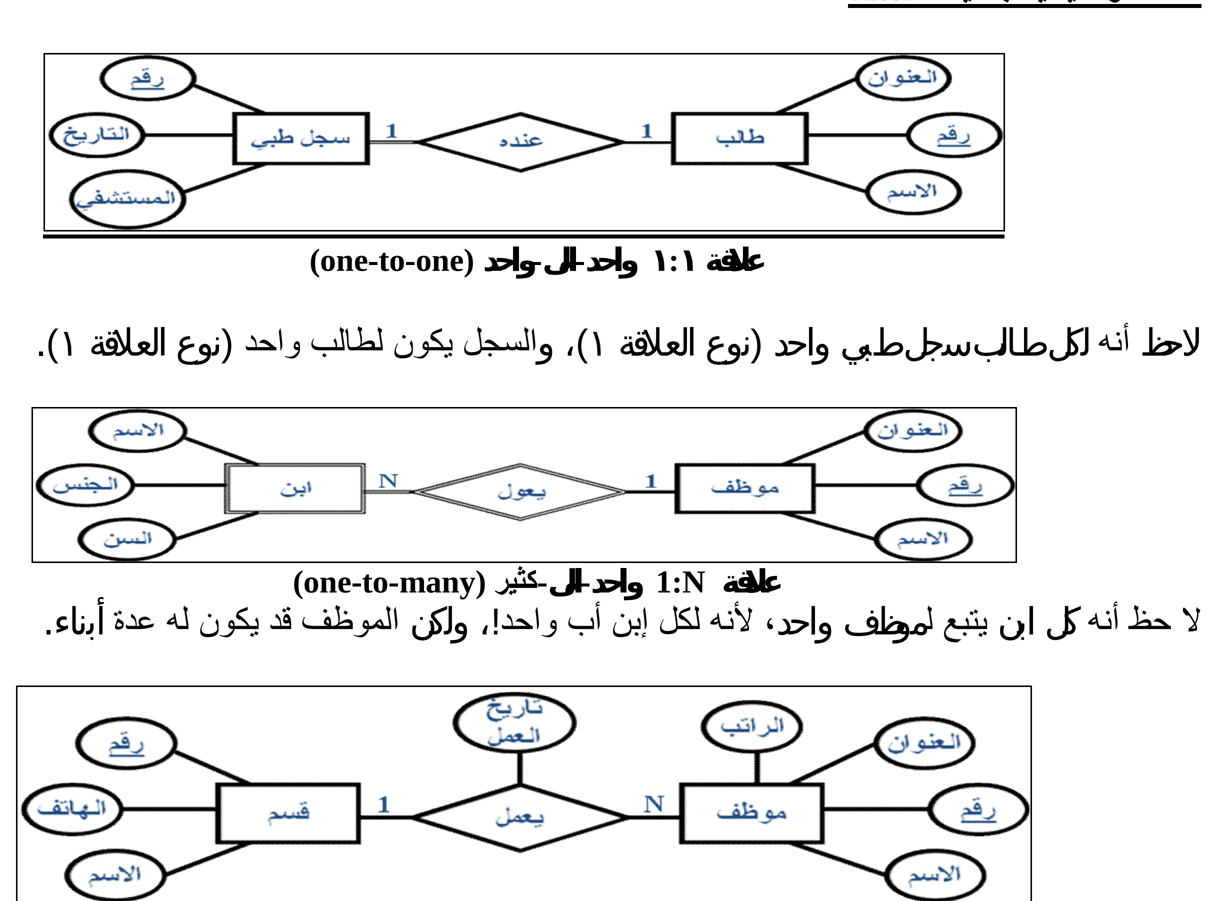
علاقة واحد-إلى-واحد (one-to-one):

وفيها ترتبط وحدة واحدة من الكيان الأول بوحدة واحدة من الكيان الآخر على الأكثر، ويرمز لها بالرمز: ١:١ .

علاقة واحد-إلى-كثير (one-to-many) :
وفيها يمكن أن ترتبط وحدة واحدة من أحد الكيانات بأكثر من وحدة في الكيان الآخر، والعكس غير صحيح، ويرمز لها بالرمز 1:N .

علاقة كثير-إلى-كثير (many-to-many) :
وفيها يمكن أن ترتبط أكثر من وحدة من الكيان الأول بأكثر من وحدة في الكيان الآخر، والعكس، أي يمكن لأي وحدة في الكيان الآخر أن ترتبط بأي وحدة في الكيان الأول، ويرمز لها بالرمز M:N.

<u>أمثلة توضيحية بصيغة ERD:</u>

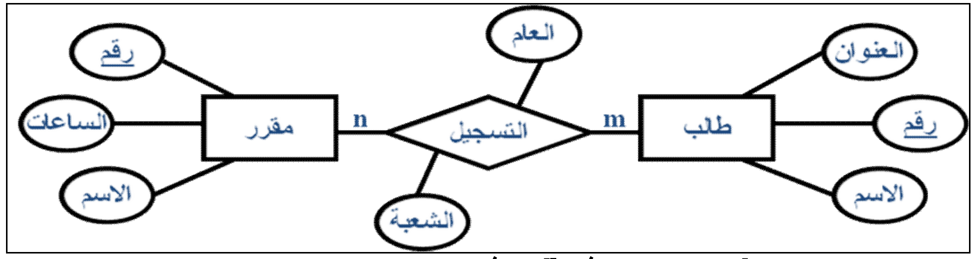


علاقة ١:١ واحد-إلى-واحد (one-to-one)

لاحظ أنه لكل طالب سجل طبي واحد (نوع العلاقة ١)، والسجل يكون لطالب واحد (نوع العلاقة ١).

علاقة 1:N واحد-إلى-كثير (one-to-many)
لا حظ أنه كل ابن يتبع لموظف واحد، لأنه لكل إبن أب واحدا!، ولكن الموظف قد يكون له عدة أبناء.

علاقة 1:N واحد-إلى-كثير (one-to-many)
لاحظ أنه لكل موظف قسم واحد، فالموظف لا يمكن أن ينتمي لأكثر من قسم إداري واحد، ولكن القسم قد يكون فيه عدة موظفين.



علاقة M:N كثير-إلى-كثير (many-to-many)
لاحظ أن الطالب قد يكون له عدة مقررات، وكذلك المقرر يمكن أن يسجله عدة طلبة.

أنواع القيود على العلاقات Relationship Constraints

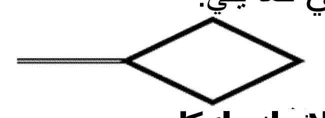
كما أوضحنا سابقا فالعلاقات تختلف فيما بينها، وكذلك شروط العلاقة تختلف حسب طبيعة العلاقة، وحسب الكيانات المرتبطة، وقيد العلاقة هو القيد الذي يعتمد على طبيعة اشتراك كيانين في علاقة ما، هل هو اشتراك كلي أم جزئي؟، ويحدد **نوع الاشتراك (Participation Constraint)** ما إذا كان وجود الكيان يعتمد على كونه مرتبط بكيان آخر عن طريق العلاقة.

أنواع قيود الاشتراك (Participation constraints):

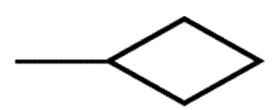
١. اشتراك كلي (Total participation).
٢. اشتراك جزئي (Partial participation).

الاشتراك الكلي (Total participation):

نقول أن العلاقة علاقة الاشتراك الكلي ، إذا كان كل وحدة في الكيان الأول يجب أن ترتبط بوحدة (Instance) من الكيان الآخر ضمن **العلاقة**، يسمي هذا القيد بقيد "**ارتباط الوجود**" (Existence Dependency)، أي أن وجود وحدة من كيان ما يستلزم ارتباطها بوحدة من كيان آخر،ومثال ذلك كيان **(سجل طبي) لطالب في نظام معلومات الجامعة**، إذ لا بد أن يرتبط أي قيد في السجل الطبي بطالب ما في كيان الطلاب، وإلا فلا معنى للعلاقة هذه.
ويتم تمثيل قيد الاشتراك الكلي، **برسم خط مزدوج**، يربط الكيانات المرتبطة بهذه العلاقة، من جهة الكيان المعتمد على **الاشتراك الكلي** كما يلي:



شكل علاقة اشتراك كلي



شكل علاقة اشتراك جزئي

اشتراك جزئي (Partial participation):

نقول أن العلاقة علاقة اشتراك جزئي، إذا كانت بعض الوحدات في الكيان المشترك بالعلاقة ترتبط ببعض الوحدات (Instances) في الكيان الآخر ضمن **العلاقة**، ويتم تمثيل قيد الاشتراك الجزئي برسم خط مفرد يربط الكيانات المرتبطة

مثال :

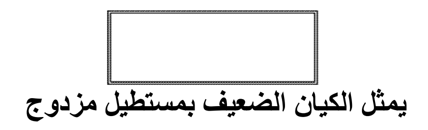
علاقة طالب بسجله الطبي في نظام معلومات الجامعة، تمثل لنا النوعين السابقين، فهي من جهة الطالب علاقة اشتراك جزئي، لأنه يمكن أن يكون هناك طالب بدون سجل، أي غير مشترك بالعلاقة، ولكنها من جهة السجل الطبي علاقة اشتراك كلي، لأنه لا يصح أن يوجد سجل طبي إلا إذا كان مرتبطا بالعلاقة مع طالب معين.

شكل هذه العلاقة تشمل على نوعي قيود الاشتراك **(الجزئي والكلي)، لاحظ أن لكل طالب سجل واحد (نوع العلاقة ١)، و**السجل يكون **لطالب واحد (نوع العلاقة ١)**، وأنه يمكن أن يكون بعض الطلبة ليس لديهم سجلات **(اشتراك جزئي)، ولكن** كل سجل لابد وأن يكون يتبع طالب معين **(اشتراك كلي).**

الكيان الضعيف (Weak Entity)

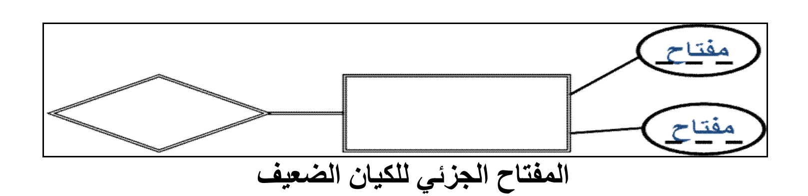
الكيانات التي لا تحتوي علي صفات تمثل مفتاح لذاتها تسمى كيانات ضعيفة, الكيان الضعيف يرتبط بكيان أخر معرف له عن طريق علاقة معرفة لهذا الكيان, **و**الكيان الضعيف دائما يرتبط بارتباط كلي مع العلاقة المعرفة له، **ولكن** ليس بالضرورة أن يكون أي كيان يشترك بالارتباط أو الاشتراك الكلي هو كيان ضعيف.

يتم تمثيل الكيان الضعيف باستخدام مستطيل مزدوج الخط، كما هو مبين بالشكل٢٦.



يمثل الكيان الضعيف بمستطيل مزدوج

يحتوي الكيان الضعيف (Weak Entity)، عادة، على مفتاح يسمي المفتاح الجزئي (Partial key)، والذي يتكون من مجموعة من الصفات التي تعرف الكيان الضعيف المرتبط بالكيان المعرف له بطريقة وحيدة، والمفتاح الجزئي يتم تمثيله عن طريق وضح خط متقطع تحت الصفات المكونة له، حيث أن هذه الصفات بمجموعها هي مفتاح الكيان entity key، ويتم تمثيل الكيان الضعيف والعلاقة المعرفة له، بخط مزدوج للأشكال المعرفة لهم في نموذج الكينونة/العلاقة.

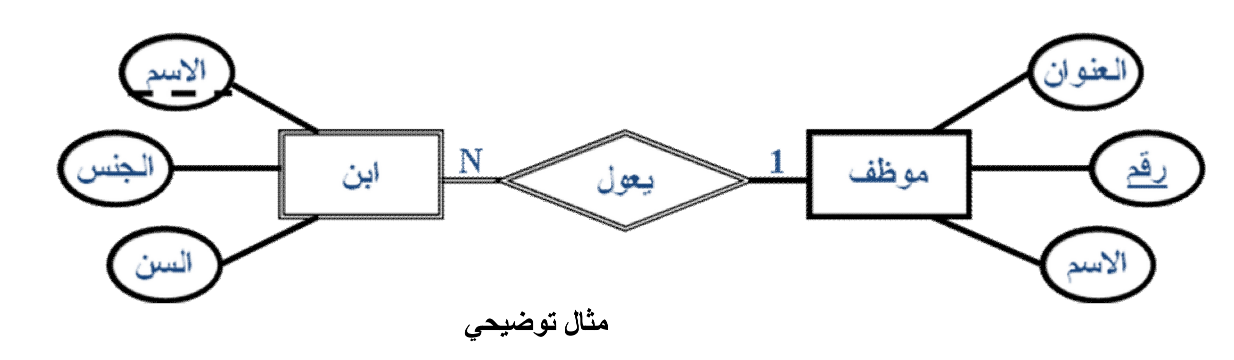


المفتاح الجزئي للكيان الضعيف

مثال :

لاحظ في الشكل ٢٧ أن:

"**أبناء**" كيان ضعيف حيث أنه لا يحتوى على مفتاح لذاته (**المفتاح الجزئي** اسم الابن يمكن ان يتكرر لموظف أخر ولكن لا يتكرر لنفس الموظف)، **العلاقة** "يعول" هي العلاقة المعرفة للكيان الضعيف "**أبناء**"، الكيان "**موظف**" هو الكيان المعرف للكيان الضعيف "**أبناء**"، الكيان الضعيف "**أبناء**" يشترك اشتراك كلي مع العلاقة "يعول".



مثال توضيحي

تمارين: اقرأ الحالات الدراسية التالية بعناية، ثم قم برسم مخطط كينونة-علاقة ERD الخاص بكل حالة

تمرين رقم ١

شركة تجارية، لديها مجموعة من **الأقسام** لتنفيذ **أعمال الشركة، ولكل قسم** البيانات:(**اسم القسم – رقم القسم** – هاتف **القسم**)، **ولدى** الشركة عدد من الموظفين، الذين يعملون **في** الأقسام الإدارية **المختلفة**، وبياناتهم: (**اسم الموظف** – الرقم الوظيفي – **العنوان – الراتب**). ويتم تسجيل تاريخ **بدء** عمل كل موظف في قسمه في سجل توظيف **الموظفين**.

اقترح نموذج بيانات لتمثيل بيانات هذه الشركة، موضحا نوع العلاقة ونوع قيود الاشتراك.

مساعدة لتوضيح أسلوب الحل:

أولا: الكيانات هي:

القسم (**اسم القسم** – رقم القسم – هاتف)

الموظف (**اسم الموظف** – الرقم الوظيفي – **العنوان – الراتب**).

ثانيا: تحديد العلاقات:

علاقة عمل الموظفون في الأقسام.

ثالثا: **مخطط** ERD؟، رابعا: **نوع العلاقات في المخطط**، خامسا: نوع قيود الاشتراك.

تمرين رقم ٢

جامعة تقوم بتدريس مجموعة من المقررات الدراسية للطلبة وقررت تسجيل البيانات الخاصة بالطلبة والمقررات وعملية تسجيل الطلبة للمقررات. كل مقرر له البيانات التالية: اسم المقرر – رقم المقرر – عدد الساعات. وبيانات الطالب المطلوبة هي: اسم الطالب – الرقم الجامعي – العنوان. يقوم مسجل الكلية بتدوين العام الجامعي والفصل الدراسي ورقم الشعبة عند تسجيل الطالب لأي مقرر.

اقترح نموذج بيانات لتمثيل بيانات هذه الجامعة، موضحا نوع العلاقة ونوع قيود الاشتراك.

تمرين رقم ٣

مركز صيانة أجهزة كهربية يقوم باستقبال أجهزة العملاء وتسجيلها في سجل الإصلاحات. ثم يقوم بتحديد الفني الذي يقوم بإصلاح الجهاز وتسجيل قطع الغيار المطلوبة للجهاز والتي طلبها الفني من المخازن. فإذا كانت بيانات الجهاز هي رقمه المسلسل و اسم الصنف وبيانات العملاء هي رقمه واسمه ورقم الهاتف. والبيانات الخاصة بالفني هي رقمه واسمه وتخصصه وبيانات قطع الغيار هي رقم القطعة و سعرها ويتم تسجيل تاريخ دخول الجهاز لمركز الصيانة في سجل الإصلاحات و يتم تسجيل عدد قطع الغيار في سجل احتياجات الأجهزة.

اقترح نموذج بيانات لتمثيل بيانات مركز الصيانة، موضحا نوع العلاقة ونوع قيود الاشتراك.

تمرين رقم ٤

مركز طبي يقوم باستقبال المرضي ويقوم الطبيب المختص بالكشف على المريض ومن الممكن أن يطلب الطبيب بعض التحاليل للمريض. بيانات الأطباء هي الاسم والرقم الوظيفي والتخصص بينما يتم تسجيل هذه البيانات للمريض: الاسم ورقم المريض، وعند كل زيارة يتم تسجيل تاريخ الزيارة. وبيانات التحليل هي اسم التحليل ورقمه وتكلفته وأيضا يتم تسجيل تاريخ إجراء التحليل. ويحتوي المركز على أماكن لرقود المرضى، ويتم تسجيل تاريخ الدخول والخروج ومسلسل التسجيل ورقم الغرفة.

اقترح نموذج بيانات لتمثيل بيانات المركز الطبي، موضحا نوع العلاقة ونوع قيود الاشتراك.

٤. دورة حياة قاعدة البيانات (DBLC) Database Life Cycle:

إن عملية تطوير قاعدة البيانات تمر بمجموعة من المراحل، هذه المراحل المتتالية تسمى بدورة حياة قاعدة البيانات، وهذه المراحل أو دورة الحياة تمر بصورة متزامنة ضمن مراحل دورة حياة نظام المعلومات، كما يوضح الشكل ٢٨.

تتكون دورة حياة قاعدة البيانات من المراحل التالية:

١) **تحديد المواصفات والمتطلبات** الخاصة بقاعدة البيانات، وهي مرحلة جزئية ضمن جمع مواصفات ومتطلبات نظام المعلومات في مرحلة التحليل.

٢) **إعداد قاعدة البيانات الأولية Conceptual Database**: وفيها يتم تصميم نموذج اولي للبيانات بواسطة مخططات **علاقة**-كينونة (E-RD).

٣) تصميم **قاعدة البيانات المنطقية Logical Database**: تحويل قاعدة البيانات الأولية، أو مخطط ERD، **إلى مخطط** الاسكيما DB Schema، وذلك بإتباع قواعد التحويل Mapping Rules.

٤) **تحسين قاعدة البيانات المنطقية**، وذلك بتطبيق قواعد تطبيع البيانات Normalization التي تهدف إلى تقليل تكرارية البيانات، من أجل رفع كفاءة قاعدة البيانات ما أمكن.

٥) تنفيذ **قاعدة البيانات الفيزيائية physical database**: وفي هذه المرحلة يتم كتابة أكواد إنشاء قاعدة البيانات بلغة SQL، ويحدد فيها بنية الجداول ونوع بيانات الحقول والمفاتيح الاساسية والاجنبية وباقي شروط تصميم قاعدة البيانات، ثم تنفيذ ذلك ضمن مدير قاعدة بيانات DBMS **مناسب، مثل** (oracle, access, sqlserver, mysql .... etc).

واعتمادا على دورة حياة قاعدة البيانات السابقة، نجد أن مرحلة التصميم تبدأ بالمرحلة الثالثة من دورة حياة قاعدة البيانات.

دورة حياة تطوير النظام SDLC	
مرحلة التخطيط	
مرحلة التحليل	
دورة حياة قاعدة البيانات DBLC	
تحديد متطلبات قاعدة البيانات	مرحلة التحليل
قاعدة البيانات الأولية ERD	مرحلة التحليل
قاعدة البيانات المنطقية LDB DB SCHEMA	مرحلة التصميم
قاعدة البيانات المنطقية المحسنة DB after NORMALIZATION	مرحلة التصميم
قاعدة البيانات الفيزيائية PDB Physical Database	مرحلة التنفيذ
مرحلة الصيانة والدعم	
دورة حياة تطوير النظام SDLC	

علاقة دورة حياة قاعدة البيانات بدورة حياة تطوير النظام عموما

تحويل مخطط كينونة – علاقة إلى مخطط قاعدة البيانات Mapping ERD to DB schema:

تتم عملية تحويل مخططERD، بتطبيق مجموعة من الخطوات البسيطة، تسمى خوارزمية التحويل Mapping Algorithm، وتتكون هذه الخطوات من جميع الحالات البسيطة المحتملة، التي قد تكون موجودة في النموذج الأولى conceptual model، ويتم تطبيق هذه الخوارزمية كاملة، مع تجاوز الحالات التي لم تظهر في النموذج الأولي،

بنهاية تطبيق هذه الخوارزمية يجب أن نحصل على مخطط قاعدة البيانات العلائقي، والذي يمثل بصورة رسومية كما سيأتي في المثال التطبيقي.

وقبل الدخول في تفاصيل خوارزمية التحويل، وقبل الحصول على مخطط قاعدة البيانات (السكيما) DB schema، سوف نقدم مجموعة من التعريفات الهامة، والتي تستخدم دائما عند وصف مخطط السكيما، وسوف نوضح عملية التحويل بمثال تطبيقي في نهاية الفصل.

العلاقة (جدول السكيما) relation:

هي علاقة رياضية ناتجة من تطبيق عملية الضرب بين مجموعتين، أو ما يسمى بالجداء الديكارتي Cartesian product، ومن هنا جاءت كلمة علاقة relation، **التي تختلف عن العلاقة** relationship المعرفة سابقا، لتعريف **التشارك** الذي يربط بين الكيانات entities، في النموذج الأولي، فيما يلي سوف نترجم الكلمة علاقة relation **إلى جدول السكيما**، وحتى نسهل الأمر، سوف نختصرها إلى **جدول**، والتي تستخدم في مرحلة تصميم قاعدة البيانات الفيزيائية، نظرا لكون جدول السكيما هو نفسه الجدول المستخدم عند تنفيذ قاعدة البيانات الفيزيائية، هذه الجداول هي مكونات مخطط قاعدة البيانات DB schema، الناتجة من إجراء عمليات تحويل مخطط كينونة-**علاقة**.

الحقل field:

هو العمود column الذي يشكل جزء من مكونات الجدول، ويتكون من مجموعة من الأعمدة أو الحقول التي تتميز **بتجانس** بيانات كل حقل، على أنه يمكن أن يكون نوع بيانات كل حقل مختلفا عن بيانات النوع الآخر.

السجل record:

هو الصف row الذي يمثل وحدة instance من وحدات الكيان، بعد تحويله إلى جدول، ويتكون الصف من الخلايا الناتجة عن تقاطعه مع الأعمدة المكونة للجدول.

المفتاح الرئيسي primary key(PK):

هو حقل في جدول يتميز بأن قيمه وحيدة في جميع صفوف الجدول، وتكون قيمته مميزة لكل صف عن أي صف آخر.

المفتاح الأجنبي foreign key(FK):

هو حقل موجود في جدول وهو لا يمثل واحدة من صفاته، ولكنه يعتبر مفتاح أجنبيا لأنه يمثل جدولا آخر، ويجب أن يكون هو نفسه المفتاح الرئيسي في ذلك الجدول، أو على الأقل تكون قيمته وحيده unique value، ويقوم المفتاح الأجنبي بتمثيل العلاقة relationship، والربط بين جدولين.

وكما سنرى لاحقا يتم التمثيل الرسومي للعلاقة relationship بين جدولين، التي كانت تمثل بشكل معين في مخطط ER، بواسطة خط طويل في مخطط قاعدة البيانات DB schema، ينطلق هذا الخط، من حقل المفتاح الأجنبي في جدول، إلى حقل المفتاح الرئيسي في جدول آخر.

خوارزمية التحويل Mapping Algorithm:

١. تحويل الكيانات العادية (القوية).	Step 1: Mapping of Regular Ent y Types
٢. تحويل الكيانات الضعيفة.	Step 2: Mapping of Weak Ent y Types
٣. تحويل العلاقات الثنائية من النوع ١:١.	Step 3: Mapping of Binary 1:1 Relaon Types
٤. تحويل العلاقات الثنائية من النوع ١:N.	Step 4: Mapping of Binary 1:N Relaonshi p Types.
٥. تحويل العلاقات الثنائية من النوع M:N.	Step5: Mapping of Binary M:N Relaonshi p Types.
٦. تحويل الصفات متعددة القيم.	Step 6: Mapping of Mulval ued ar i but es.
٧. تحويل العلاقات فوق الثنائية.	Step 7: Mapping of N-ary Relationship Types.

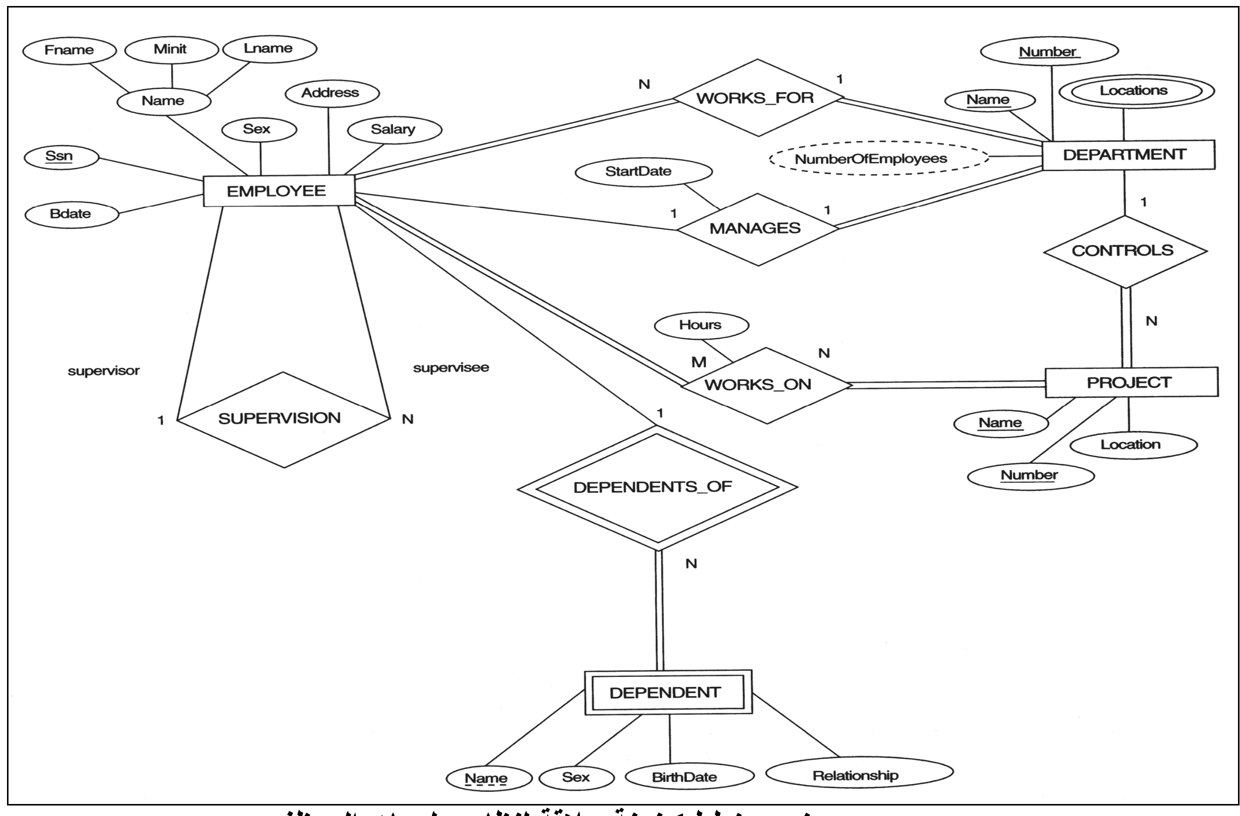
١. تحويل أنواع الكيانات العادية:

يتم هنا تحويل جميع الكيانات العادية، أي الكيانات غير الضعيفة، بإنشاء جدول يتكون من الحقول التي تقابل صفات ذلك الكيان. ويتم تحديد أحد مفاتيح الكيان، وتسميته بالمفتاح الرئيسي primary key(PK)، **وإذا كانت الصفة التي تمثل المفتاح من النوع المركب** (composite attributes)، **فإن المفتاح** الرئيسي سيكون مجموعة الحقول التي تنشأ من **الصفة المركبة**.

٢. تحويل الكيانات الضعيفة:
يتم تحويل كل واحدة من الكيانات الضعيفة، بإنشاء جدول يتكون من الحقول التي تقابل صفات ذلك الكيان، كما يجب إضافة المفتاح الرئيسي للكيان القوي الذي يتبعه ذلك الكيان الضعيف، ويكون المفتاح الرئيسي PK للجدول الجديد، عبارة عن مفتاح مركب مكون من المفتاح الأجنبي FK، بالإضافة إلى المفتاح الجزئي (Partial Key) الخاص به.
٣. تحويل العلاقات الثنائية من النوع ١:١:
إذا كانت العلاقة بين الكيانين علاقة واحد-إلى-واحد (one-to-on)، فإن عملية التحويل تتم وفق عدة خيارات أشهرها، خيار يسمى **بطريقة المفتاح الأجنبي**، وفيه يتم إضافة المفتاح الرئيسيPK لأحد الجدولين إلى الجدول الآخر كمفتاح أجنبيFK، ويفضل أن يكون الجدول الذي يحتوي على المفتاح الأجنبي، هو الجدول الذي يكون نوع قيد اشتراكه في العلاقة، من نوع (الاشتراك الكليTotal Participation)، والمثال التطبيقي يوضح ذلك.
كما أن هناك خيار آخر، يحدث غالبا عندما يكون كلا الكيانين مشتركين بقيد الاشتراك الكلي، حيث يتم دمج الكيانين في جدول واحد، وتعيين مفتاح رئيسي جديد، أو مفتاح رئيسي مركب مكون من كلا المفتاحين الرئيسيين.
٤. تحويل العلاقات الثنائية من النوع ١:N:
يتم هنا إنشاء جدولين لتمثيل الكيانين المرتبطين، على أن يتم تطبيق **طريقة المفتاح الأجنبي** السابقة، وذلك بإضافة المفتاح الرئيسيPK للجدول من جهة العلاقة (N)، إلى الجدول الآخر المرتبط بالعلاقة (١)، بغض النظر عن نوع قيد الاشتراك.
٥. تحويل العلاقات الثنائية من النوع N:M:
في هذا النوع من العلاقات، يتم استحداث جدول جديد، فيكون الناتج من هذه العلاقة ثلاثة جداول، جدولين لتمثيل الكيانين المرتبطين بالعلاقة relationship، ويضم الجدول الثالث حقلين كمفتاحين أجنبيين يمثلان المفتاحين الرئيسيين في الجدولين، ويمكن إضافة أي حقل آخر يكون له مغزى، كأن تكون العلاقة لها صفة بذاتها، فتتحول الصفة إلى حقل في الجدول الجديد.
٦. تحويل الصفات متعددة القيم:
يتم في هذه الحالة، عادة، إنشاء جدول جديد يضم الصفة المتعددة القيم كحقل، ويضاف إلى الجدول مفتاح أجنبيFK، يكون ممثلا للمفتاح الرئيسي في الجدول الناتج من الكيان الذي يحتوي على الصفة متعددة القيم.
أما الصفات المركبة فتتحول إلى صفات بسيط، فحقول عادية كما أوضحنا أعلاه، والصفات ذات القيم المشتقة تلغى من الجدول، لأنها صفات قابلها للاشتقاق من صفات أخرى، فلا داعي لوجودها.
٧. تحويل العلاقات غير الثنائية، كالعلاقة الثلاثية وما فوقها:
في حالات نادرة تظهر لدينا علاقات معقدة، كالعلاقة الثلاثية (بين ثلاثة كيانات) والرباعية وما فوقها، وتعالج هذه الحالة بطريقة معالجة الحالة الخامسة (حالة تحويل العلاقات الثنائية من النوع N:M)، حيث يتم إنشاء جدول جديد، وإضافة المفاتيح الرئيسية للجداول المشتركة، حسب عددها، إلى الجدول الجديد كمفاتيح أجنبية مكونة بمجموعها، مفتاحا مركبا يمثل المفتاح الرئيسي للجدول.

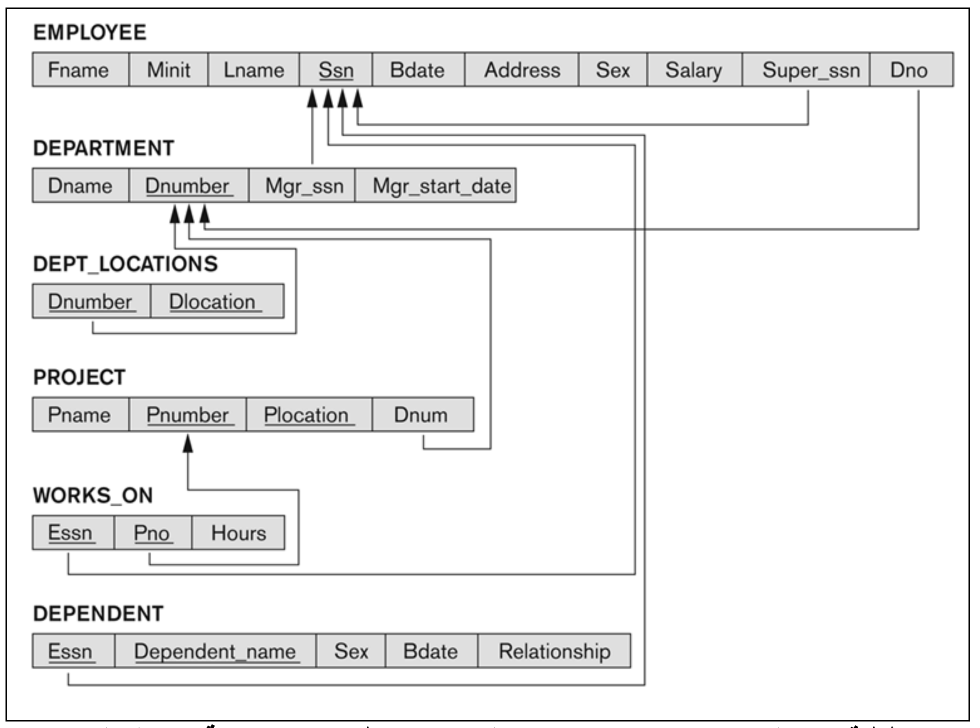
مثالين توضيحيين:
يوضح المثل الأول الحالات الست الأولى، ويوضح الثاني الحالة الأخيرة بمثال عن العلاقة الثلاثية.
المثل الأول: (قاعدة بيانات موظفين employee)
يتكون مخطط ERD التالي من الكيانات والعلاقات الموضحة في الجدول التالي، والذي يسمى بقاموس قاعدة البيانات، إذ اضفنا له تعريفات الحقول وأنواع بياناتها، كما يجب أن تنفذ في قاعدة البيانات الفيزيائية.

الكيان أو العلاقة	**Entity/relationship**	**الوصـــف**
موظف	Employee	كيان يمثل موظفا في النظام.
قسم	Department	كيان يمثل قسما إداريا.
مشروع	Project	كيان يمثل مشروعا يعمل عليه الموظف حتى ينتهي.
عائلة	Dependent	كيان يمثل أفراد عائلة الموظف الذين يعولهم.
يعمل لـ	Works_for	علاقة ثنائية نوعها ١:N، كل الموظف يعمل لقسم إداري معين.
يدير	Manages	علاقة ثنائية نوعها ١:١، الموظف الذي وظيفته مدير، يدير قسما إداريا معينا.
يتحكم	Controls	علاقة ثنائية نوعها ١:N، القسم الإداري يتحكم بعدة مشاريع.
يعول	Dependents_of	علاقة ثنائية نوعها ١:N، كل موظف يعول عدة أفراد من أسرته.
يعمل على	Works_on	علاقة ثنائية نوعها M:N ، كل موظف يعمل على أكثر من مشروع.
يشرف	Supervision	علاقة ثنائية نوعها ١:N، يوجد موظف يشرف على عدة موظفين، كمدير لهم. **وهذا النوع من العلاقات يطلق عليه (العلاقة التعاودية Recursive relation) حيث أنها علاقة بين وحدات من نفس الكيان، فالمدير هنا هو موظف في نفس الوقت، ويرأس موظفا يمثل بوحدة ضمن نفس الكيان.**



يوضح مخطط كينونة-علاقة لنظام معلومات الموظفين

بعد تطبيق خوارزمية تحويل المخطط السابق ينتج لدينا مخطط قاعدة البيانات (DB schema)، **الموضح في الشكل٣٠.**

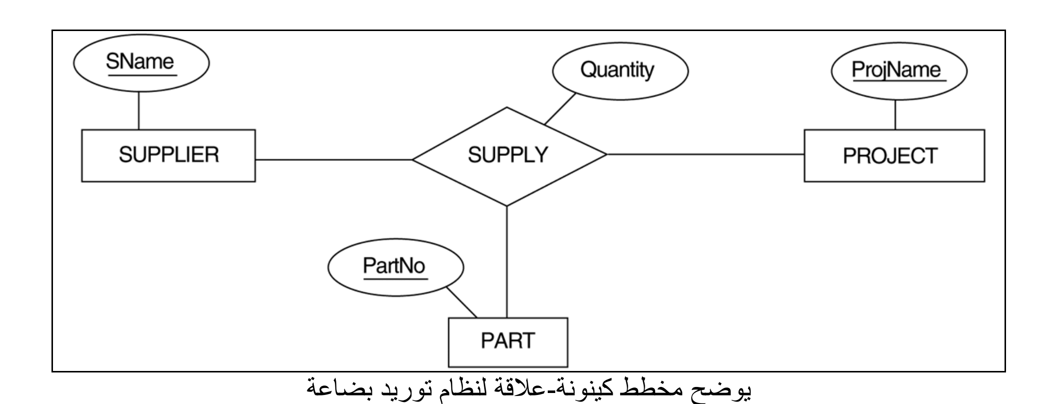


مخطط قاعدة البيانات (DB schema) الناتج من تطبيق خوارزمية تحويل الـERD

المثل الثاني: (مخطط ER يحتوي على علاقة ثلاثية)

يتكون مخطط كينونة-علاقة أدناه من الكيانات والعلاقات الموضحة في الجدول التالي:

الكيان أو العلاقة	Entity/relationship	الوصف
مورد	Supplier	كيان يمثل بيانات المورد.
صف أو قطعة	Part	كيان يمثل بيانات صنف أو قطعة من البضاعة الموردة.
مشروع التوريد	Project	كيان يمثل مشروع عملية التوريد.
توريد	Supply	علاقة ثلاثية (بغض الظر عن نوعها) تشترك فيها الكيانات الثلاثة.



يوضح مخطط كينونة-علاقة لنظام توريد بضاعة

وبالطبع فإن تحويل مخطط ER أعلاه، يؤدي بنا إلى مخطط قاعدة البيانات التالية في الشكل ٣٢.

SUPPLIER

SNAME	. . .

PROJECT

PROJNAME	. . .

PART

PARTNO	. . .

SUPPLY

SNAME	PROJNAME	PARTNO	QUANTITY

مخطط قاعدة البيانات (DB schema) الناتج من تطبيق خوارزمية تحويل الـERD

على الطالب القيام بتوصيل الأسهم التي توضح المفاتيح الرئيسية والأجنبية والعلاقات بين الجداول في مخطط السكيما هذا.

٥. قواعد التطبيع Normalization rules:

هي عملية تقسيم البيانات إلى مجموعة من العلاقات التي تتسم بالتجانس والاعتمادية المتبادلة والخالية من التكرار. وبالطبع فإن القضاء على التكرارية تماما أمر غير ممكن، وما يتم دائما هو تقليل التكرارية ما أمكن كما سيأتي، إن قواعد التطبيع هي مجموعة المعايير المتبعة في تصميم قواعد البيانات، وهي خطوات لتجزئة العلاقات relations (الجداول)، إلى علاقات أصغر.

ورغم اهمية عملية نمذجة البيانات في مخطط كينونة علاقة، أو ما يسمى بقاعدة البيانات الأولية، فإن تطبيع البيانات أو قواعد تسوية البيانات لا تقل أهمية عنها في نظرية قواعد البيانات. إن تطبيع البيانات كما هو واضح من الاسم هو عملية جعل البيانات طبيعية أكثر مقارنة بالنموذج الرياضي المثالي.

لماذا نقوم بتطبيع البيانات ?Why Normalization:

- لمعرفة وفهم المعنى الكامن في البيانات التي تحتويها قاعدة البيانات.
- لمعرفة وفهم العلاقات والإعتماديات المتبادلة بين البيانات، ويقصد بالإعتماديات المتبادلة كون معالجة بيانات معينة (إدخال تعديل أو حذف) تعتمد على معالجة بيانات أخرى.
- لتقليل تكرارية البيانات (data redundancy) والتناقض الذي قد يحصل بينها.
- لحماية البيانات من المشاكل التي قد تطرأ عند معالجتها، أي عند إجراء إدخال بيانات جديدة أو تعديل بيانات موجودة أو حذفها.

مميزات تطبيق قواعد التطبيع Normalization Advantages :

يمكن إجابة السؤال السابق عن طريق إجابة سؤال آخر، ما هي مميزات تطبيق قواعد التطبيع؟، كما يلي:

- تقليل تكرارية البيانات من خلال تصميم جداول قاعدة البيانات الفيزيائية، وذلك يؤدي إلى تقليل مساحة التخزين على القرص الصلب.
- تقليل التضارب الحاصل بين البيانات.(التضارب هو عدم التجانس inconsistency)
- تقليل تكلفة تحديث البيانات ومعالجتها.
- إلغاء العلاقات التي من نوع many-to-many، وذلك بتحويلها إلى علاقات من نوع one-to-many .
- زيادة وتطوير مرونة النظام.

عيوب تطبيع البيانات:

و كأي شيء في الحياة، فكما أن لتطبيق قواعد التطبيع مميزات، فإن له عيوبا منها:

- الاضطرار (أحيانا) إلى استخدام الفهارس التي تعقد عملية المعالجة وإن كانت ترفع من كفاءتها.
- زيادة تعقيدات النظام.
- تقليل كفاءة بعض أوامر استرجاع البيانات، خاصة العلاقات المعقدة، التي يتم عمل ارتباطات كثيرة لها عند الاسترجاع.

وبسبب هذه العيوب، يتم في بعض الأحيان، تطبيق قواعد التطبيع العكسية، أو ما يسمى Demoralization.

تمرين للبحث: ما هي قواعد عكس التطبيع Demoralization؟وضح بمثال؟

الاعتمادية الوظيفية Functional Dependencies:

هي وجود قيد بين صفتين two attributes (عمودين)، أو بين مجموعتين من الصفات.

فعندما نقول أن العمود B يعتمد وظيفيا على العمود A **في نفس الجدول فإننا نرمز لذلك بـ:**

A ⟶ B (**العمود** A يحدد العمود B، **أو العمود** B يعتمد على A)

في المثال التالي، الجدول student يحتوي على الحقول أو الأعمدة الأربعة التي تعتمد على العمود studentID، كما هو موضح أدناه.

studentID	studentName	Major	College	Loc

العمود studentID يحدد مجموعة الأعمدة كما هو موضح بالصيغة:

studentID ⟶ { studentName,Major,College,Log }

وبطبيعة الحال يمكن كتابة الجملة السابقة بطريقة اعتمادات وظيفية منفصلة، بحيث نشير إلى أن رقم الطالب يحدد اسم الطالب، وهكذا..

اللاعتمادية الوظيفية الجزئية:

عندما يكون هناك عمود معتمد وظيفيا على مفتاح رئيسي مركب، فإن الاعتمادية الجزئية تعني أن العمود معتمد على جزء من الفتاح الرئيسي، وليس كله.

ويعبر عن ذلك كالتالي:

إذا كان العمود A مفتاحا رئيسيا مركبا، بحيث أن A=BC، وكان لدينا العمود D بحيث أن:

A→D، **فإننا نقول أن** D يعتمد جزئيا على A، **إذا كان العمود** C فقط هو الذي يحدد D.

ونقول أن العمود D يعتمد كليا على A، إذا كان كلا من العمودين B و C يحددان D **معا**.

الاعتمادية الوظيفية الكلية أو الكاملة:

تكون هناك اعتمادية كاملة في حالتين:

- **كان المفتاح** الذي يحدد الأعمدة الأخرى غير مركبا، أي مفردا.
- أو كان المفتاح مركبا، وكل مكوناته تحدد الأعمدة الأخرى معا.

الاعتمادية الوظيفية المتعدية:

إذا كان العمود A يحدد العمود B **وكان العمود** B يحدد C، **فإننا نسمي الجملة** A→C إعتمادية متعدية، لأنها نتجت عن **الجملة** A→B، و B→C.

الأشكال الطبيعية Normal Forms:

هي حالات أو حالة خاصة بالعلاقة relation **(الجدول)**، والتي تنتج من تطبيق قواعد بسيطة متعلقة بالاعتمادية الوظيفية على تلك العلاقة.

وقواعد التطبيع هي مجموعة من الشروط المتتالية، بحيث لا يعتبر الثاني محققا إلا بعد تحقق الأول، والثالث يعتمد في تحققه على الثاني، أي على تحقق الأول والثاني معا، وهكذا، تقدم هذه الشروط مجموعة خصائص يفترض تحققها في كل قاعدة، وهذه القواعد هي:

- 1st Normal Form (1NF)
- 2nd Normal Form(2NF)
- 3rd Normal Form(3NF)
- Boyce-Codd Normal Form(BCNF)
- 4th Normal Form(4NF)
- 5th Normal Form(5NF)

القاعدة الأولى - الشكل الطبيعي الأول first normal form (1NF):

تشترط هذه القاعدة من أجل وصول العلاقة **(أو الجدول)** إلى الصيغة أو الشكل الطبيعي الأول، منع وجود القيم المتعددة **في حقل** معين ومنع وجود القيم المركبة أو القيم المشتقة في أي حقل، وبالتالي يجب أن تكون القيم في مجال الحقل قيماً ذرية Atomic أولية، بمعنى تكون بسيطة وغير قابلة للتفكك.

القاعدة الثانية - الشكل الطبيعي الثاني second normal form (2NF):

تشترط هذه القاعدة من أجل الوصول **الجدول (أو العلاقة)** إلى الصيغة الطبيعة الثانية:

١) **أن تكون العلاقة (الجدول)** في الشكل الطبيعي الأول.

٢) أن لا تكون هناك أي اعتمادية وظيفية جزئية، بمعنى أن أي اعتمادية وظيفية يجب أن تكون كاملة كما سيأتي.

القاعدة الثالثة - الشكل الطبيعي الثالث third normal form (3NF):

تشترط هذه القاعدة من أجل الوصول الجدول (أو العلاقة) إلى الصيغة الطبيعة الثالثة:

١) أن تكون العلاقة في الشكل الطبيعي الثاني.

٢) أن لا تكون هناك أي اعتمادية وظيفية متعدية، بحيث يتم تحويل الاعتمادية المتعدية إلى متعدية كاملة.

القواعد الأخرى:

سنكتفي فيما يلي بدراسة القواعد الثلاث الاساسية مع العلم بوجود قواعد أخرى أكثر تعقيدا هي:

قاعدة بويس- كود BCNF، وقاعدة الشكل الطبيعي الرابع، والشكل الطبيعي الخامس.

ما هي المخرجات الناتجة عن تطبيق قواعد التطبيع؟

المخرجات الناتجة هي قاعدة بيانات منطقية سليمة وخالية من العيوب، على أن تكون على الأقل في الشكل الطبيعي الثالث، وتكون حقول أو أعمدة كل جدول معتمدة كليا على عمود واحد أو عدة أعمدة تمثل المفتاح الاساسي فقط لا غير.

ملاحظة هامة:

قواعد تطبيع البيانات ليست مجرد قواعد رياضية صارمة فقط، فتطبيقها يعتمد أساسا على فهم قاعدة البيانات، وفهم طبيعة كل عمود في العلاقة وطبيعة البيانات التي تخزن فيه، وهذه شيء يعود دائما على مصمم قاعدة البيانات ومدى فهمه لمرحلة تحليل النظام.

أمثلة توضيحية حول قواعد التطبيع

١)القاعدة الأولى 1NF:

- إذا كانت العلاقة غير مطابقة للشكل الطبيعي الأول، فإننا نقول أنها في الشكل الطبيعي صفر ونرمز لذلك بـ **0NF.**
- نقوم بفحص العلاقة في الشكل الطبيعي صفر، ونقوم بتحويل العمود ذو القيم المركبة أو متعددة القيم إلى قيم بسيطة، إما بتحويل العمود الواحد إلى عدة أعمدة، أو بعمل علاقة جديدة تحوي القيم المتعددة أو المركبة، **بالإضافة** إلى مفتاح أجنبي يمثل الجدول الأصلي بالطبع.
- المثال التالي يوضح وجود الحقلين children و gender اللذين يجعلان العلاقة EMP **من الشكل** 0NF، هذا **بالإضافة إلى الحقل** Nun_children الذي يحتوي كما هو واضح على قيمة مشتقة، هي عدد الأطفال.

Emp_ID	Children	Gender	Nun_children
002	Nan, Pat, Lee	F, M, M	3
006	Todd	M	1

- **وللعودة بالعلاقة** EMP **إلى الشكل** الطبيعي الأول 1NF، نقوم بتحويل الحقلين متعددي القيم إلى أكثر من حقل، ونقوم بحذف العمود ذا القيمة المشتقة.

Remove derived attributes

Emp_ID	Ch1	G1	Ch2	G2	Ch3	G3	Ch4	G4	Nun_children
002	Nan	F	Pat	M	Lee	M			3
006	Todd	M							1

- فيصير الجدول EMP **كالتالي:**

Emp_ID	Ch1	G1	Ch2	G2	Ch3	G3	Ch4	G4
002	Nan	F	Pat	M	Lee	M		
006	Todd	M						

- وهكذا نحكم أن الجدول EMP صار في الشكل الطبيعي الأول 1NF.

٢)القاعدة الثانية 2NF:

- علينا دائما قبل التحقق من شروط الشكل الطبيعي الثاني، ان نقوم بالتأكد من تحقق الشكل الطبيعي الأول، أي أن العلاقة تحقق القاعدة 1NF.
- نهدف دائما في الشكل الطبيعي الثاني ان نحول أي اعتمادية جزئية إلى اعتمادية كاملة.
- إذا كان المفتاح الاساسي في العلاقة مكونا من عمود واحد فهذا يعيني بالضرورة عدم وجود إعتمادية جزئية.
- وفي حالة وجود مفتاح رئيسي مركب فإننا نبدأ في التأكد من عدم وجود إعتمادية جزئية كما يوضح المثال التالي.
- الجدول أدناه يمثل اسماء مجموعة من الاشخاص الذي يشترون أسهم تابعة لشركات مختلفة.

Cust_ID	Name	Addr	Stock	Curr_price	Dividend	Shares
003	Smith	LA	IBM	100	3	16
019	Jones	NY	USX	50	2	5
102	Harris	KC	IBM	100	3	10
102	Harris	KC	CVD	1	0	1000

- نلاحظ أن المفتاح الاساسي المناسب للجدول أعلاه، هو المفتاح المركب المكون من الحقلين رقم الزبون cust_ID وجهة السهم stock (الشركة التي تبيع السهم).
- وإذا قمنا بفحص الاعتمادية نلاحظ التالي:

- Cust_ID → Name,Addr (Partial FD)

- **العمود (رقم الزبون)** يحدد العمودين **(الاسم، والعنوان)** ....اعتمادية جزئية.

- Stock → Curr_price,Dividend (Partial FD)

- العمود جهة السهم يحدد العمودين **(السعر الحالي)** و **(الربح)** ... اعتمادية جزئية.

- Cust_ID,Stock → Shares (full FD)

- المفتاح الأساسي المركب من رقم الزبون وجهة السهم يحدد العمود **عدد الاسهم** .. وهذه فقط هي الاعتمادية **الكاملة**.
- وبالطبع نحذف الاعتماديات الوظيفية الجزئية محولين الجدول السابق إلى ثلاثة جداول، يحتوي الجدول الأصلي على الاعتمادية الكاملة، ويظهر جدولين بكل واحد منهما الاعتمادية الجزئية التي تصير عند فصلها اعتمادية كاملة.

Cust_ID	Stock	Shares
003	IBM	16
019	USX	5
102	IBM	10
102	CVD	1000

Cust_ID	Name	Addr
003	Smith	LA
019	Jones	NY
102	Harris	KC
~~102~~	~~Harris~~	~~KC~~

Stock	Curr_price	Dividend
IBM	100	3
USX	50	2
~~IBM~~	~~100~~	~~3~~
CVD	1	0

- وبحذف الأعمدة الجاهزة ينتج لدينا الجداول الثلاثة التالية:

Cust_ID	Stock	Shares
003	IBM	16
019	USX	5
102	IBM	10
102	CVD	1000

Cust_ID	Name	Addr
003	Smith	LA
019	Jones	NY
102	Harris	KC

Stock	Curr_price	Dividend
IBM	100	3
USX	50	2
CVD	1	0

- الجداول أو العلاقات السابقة relations في الشكل الطبيعي الثاني 2NF.

٣)القاعدة الثالثة 3NF:

- بعد أن تحقق من كون العلاقة في الشكل الطبيعي الثاني نبدأ بالتحقق من تحقق الشكل الطبيعي الثالث.
- نهدف في الشكل الطبيعي الثالث القضاء على أي إعتمادية متعدية، وتحويل العلاقة إلى الاعتمادية الكاملة، ويكون ذلك غالبا بفصل أعمدة الاعتمادية المتعدية عن العلاقة الأولى.
- علينا أن نتأكد من كون كل الاعمدة تعتمد على المفتاح الاساسي مباشرة، ولا تعتمد عليه عن طريق عمود (غير مفتاحي) آخر.
- في المثال التالي العلاقة تمثل جدولا يتكون من رقم الصنف كمفتاح اساسي، واسم الصنف والشركة المصنعة وبلد الشركة المصنعة كأعمدة غير مفتاحية.

Product#	Name	Mfr	Mfr_HQ
1001	Walkman	Sony	Japan
1002	Camera	Leica	Germany
1003	VCR	Sony	Japan

- واضح تماما أن المفتاح الاساسي يحدد اسم الشركة المصنعة، وأن اسم الشركة يحدد بلد الشركة، وعليه فإن **العلاقة المتعدية** هي كون المفتاح الاساسي **(رقم المنتج)** يحدد بلد الشركة المصنعة.

Product# ⟶ Mfr

Mfr ⟶ Mfr_HQ

Therefore Product# ⟶ Mfr_HQ

- يجب إلغاء العلاقة المتعدية بتحويل الجدول السابق إلى جدولين يتكون كل واحد من إعتمادية كاملة كما يلي:

Product#	Name	Mfr	Mfr_HQ
1001	Walkman	Sony	Japan
1002	Camera	Leica	Germany
1003	VCR	Sony	Japan



Product#	Name	Mfr
1001	Walkman	Sony
1002	Camera	Leica
1003	VCR	Sony

Mfr	Mfr_HQ
Sony	Japan
Leica	Germany

- كما هو واضح من الشكل السابق تم فصل الجدول (المنتج)، إلى جدولين هما المنتج و جدول الشركات فيه المفتاح الاساسي لسم الشركة Mfr، بحيث يكون مفتاح اجنبيا في جدول المنتج، والذي يظل المفتاح الاساسي به كما كان سابقا.

تمرين للطالب:

قم بتطبيق قواعد التطبيع normalization على الجدول التالي، الذي يظهر أنه في الشكل الطبيعي الصفري 0NF، حتى وصوله إلى الشكل الطبيعي الثالث 3NF، علما أنه يمثل قاعدة بيانات نظام تأجير أقراص مدمجة CDs.

Receipt	Date (M/D/Y)	PID	PName	Phone	City	State	VID	VName	Type	Days	Cost
68395	9/1/01	226	Lindsey Moore	(734)763-4385	ST Ann Arbor	MI	325. 548. 6437	Shrek, Remembering, The Replacement Killers	N, H, H	1, 2, 2	3.00. 2.50. 2.50
68397	9/4/01	224	Helen Baker	(743)763-2138	Chicago	MI	325. 6437	Shrek, The Replacement Killers	N, H	1, 2	3.00. 2.50
69001	10/2/01	226	Lindsey Moore	(734)763-4385	ST Ann Arbor	MI	468	True Lies	N	1	3.00

وأن قاموس بيانات الجدول كالتالي:

إسم العمود	البيان
Receipt	رقم إيصال الإستلام الذي يأخذه المستأجر معه.
Date	تأريخ عملية التأجير.
PID	رقم الزبون الذي قام بعملية الاستئجار.
Pname	اسم الزبون.
Phone	رقم هاتف الزبون.
City	العنوان ـ المدينة
State	العنوان ـ الولاية
VID	أرقام الافلام التي في الأقراص المستأجرة.
Vname	اسماء الافلام المستأجرة.
Type	نوع الفيلم أو نوع القرص؟.
Days	عدد ايام الايجار لكل فيلم.
Cost	تكلفة الاستئجار وهو المبلغ الذي على الزبون دفعه.

٦. قاعدة البيانات الفيزيائية

قاعدة البيانات الفيزيائية PDB

قاعدة البيانات الفيزيائية physical database هي قاعدة البيانات المادية، أو الفعلية، والتي تم تنفيذها في جهاز الحاسوب، ونستخدم لأجل ذلك، في الغالب، لغة SQL، ويتم تنفيذ أكوادها في نظام إدارة قاعدة البيانات DBMS الذين اخترناه، وفي الجدول التالي نوضح أسماء مجموعة من أنظمة إدارة قاعدة البيانات المتوفرة.

م	DBMS Brand	الشركة المصنعة	ملاحظات
١	Oracle	ORACLE	أقوى وأشهر برمجيات نظم إدارة قواعد البيانات الشبكية.
٢	Ms-Access	Microsoft	هام لقواعد البيانات الصغيرة والبسيطة.
٣	Ms-SqlServer	Microsoft	يقارب أوراكل مع دعم التطبيقات الخاص بميكروسوفت.
٤	Mysql	(OpenSource)	من أهم نظم إدارة قواعد بيانات الويب، مع دعم في لغة PHP المعروفة.
٥	DB2	IBM	يضارع قوة الأوراكل، مع دعم منصات عمل خادمات شركة IBM المعروفة.

جميع نظم إدارة البيانات DBMSs، تقدم للمصمم ميزة استخدام لغة الاستعلام البنيوي SQL، ضمن بيئتها، وباستخدام أنواع البيانات الخاصة بها، وتستخدم لذلك مجموعة من الايعازات التي نسميها بلغة تعريف البيانات DDL: data definition language، تساعد هذه الإيعازات على بناء الجداول والحقول وتعريف خصائصها وقيودها، وبناء أي كائنات أخرى خاصة بقاعدة البيانات، ومنها المنظور view كما سيأتي.

التعامل مع قاعدة البيانات الفيزيائية:

من أجل التعامل مع الـ PDB، نحتاج إلى لغة الإستعلامات البنيوية الشهيرة SQL، والتي تقدم لنا مجموعة من الجمل CLAUSE، باستخدام عدد كبير من الأوامر، ونشير هنا إلى أن أهم أوامر هذه اللغة تصنف حسب الاستخدام إلى عدة مجموعات رئيسية منها:

١- لغة استرجاع البيانات DRL: Data Retrieval Language.
٢- لغة معالجة البيانات DML: Data Manipulation Language.
٣- لغة تعريف البيانات DDL: Data Definition Language.
٤- لغة التحكم بالبيانات DCL: Data Control Language.

وسنهتم في هذا الفصل بلغة DDL، على أن نغطي DRL و DML و DCL في الفصول اللاحقة.

لغة تعريف البيانات DDL:

هي جزء من لغة الـ SQL التي تهتم بإنشاء وتعديل وحذف جميع كائنات قاعدة البيانات، بما فيها إنشاء وتعديل وحذف قواعد البيانات أو أي من كائناتها كالجدول والمنظور View، وعمليات الإنشاء والحذف تتعلق بتعريف كائنات قاعدة البيانات لا بياناتها نفسها، فالفارق الرئيسي بين لغة DDL ولغة DML، أن لغة DDL لغة تعريف بنية الكائن، بينما تهتم الثانية DML بمعاجلة البيانات داخل الكائن نفسه، دون أن تتدخل ببنية الكائن.

جملة create table:

نستخدم لإنشاء الجدول، الجملة create table، وهذه الجملة تستخدم حسب المثال التالي، الذي يوضح جملة إنشاء جدول إسمه قسم department ويحتوي على الحقول الموضحة في الجملة:

```
CREATE TABLE DEPARTMENT (DNAME          VARCHAR(10)     NOT NULL,
        DNUMBER         INTEGER         NOT NULL,
        MGRSSN          CHAR(9),
        MGRSTARTDATE    CHAR(9)  );
```

لاحظ أن تعريف الحقل يتكون من اسم الحقل ونوع بياناته وشرط من شروط، كما سيأتي معنا، ويمكن بالطبع ذكر اسم الحقل ونوع البيانات بدون ذكر أي شرط، وبالإضافة إلى أسماء الحقول، فإن جملة الإنشاء هذه تحتمل أن تتكون من جمل تحديد المفتاحين الرئيسي والأجنبي في الجدول، وبعض الشروط عليهما، كما يوضح المثال التالي، الذي نقوم فيه بإنشاء وتعريف الجدول dept، كما يلي:

```
CREATE TABLE DEPT (
        DNAME                   VARCHAR(10)     NOT NULL,
        DNUMBER         INTEGER         NOT NULL,
        MGRSSN          CHAR(9),
        MGRSTARTDATE    CHAR(9),
        PRIMARY KEY (DNUMBER),
        UNIQUE (DNAME),
        FOREIGN KEY (MGRSSN) REFERENCES EMP(SSN));
```

لاحظ الأسطر الثلاثة الأخيرة، حيث يتم تحديد الحقل (DNUMBER) كمفتاح أساسي، وتحديد أن الحقل (DNAME) ذو قيم فريدة، وأخيرا أن الحقل (MGRSSN) هو مفتاح أجنبي معطى من الجدول (EMP) كمقابل للحقل (SSN).
سنرى في القسم الثاني من هذا الفصل، المزيد من الشروط الذي يمكن تطبيقها على الجدول.

جملة alter table:

من أجل تعديل بنية أي جدول نستخدم الجملة alter table، والتي تسمح بإضافة أو حذف أعمدة/حقل أوشروط للجدول، المثال التالي يوضح جملة إضافة حقل جديد اسمه JOB للجدول EMPLOYEE:

```
ALTER TABLE EMPLOYEE ADD JOB VARCHAR(12);
```

ونلخص مهام هذه الجملة بالتالي:

الوظيفة	جملة التعديل
إضافة عمود جديد إلى الجدول	ALTER TABLE (tablename) ADD (columnname) ;
حذف عمود من الجدول	ALTER TABLE (tablename) DROP (columnname) ;
إضافة قيد أو شريط جديد للجدول	ALTER TABLE (tablename) ADD (constraint_name);
حذف شرط موجود من الجدول	ALTER TABLE (tablename) DROP CONSTRAINT constraint_name;
تحديد قيمة تلقائية لعمود في الجدول	ALTER TABLE (tablename) ADD (columnname) DEFAULT;
إلغاء أو حذف قيمة تلقائية من عمود في الجدول	ALTER TABLE (tablename) ALTER (columnname) DROP DEFAULT;

جملة Drop Table:

تستخدم هذه الجملة لحذف جدول بأكمله من قاعدة البيانات، وتستخدم كما يوضح المثال التالي:

```
DROP TABLE  DEPENDENT;
```

ميزة تعزيز سلامة البيانات Integrity Enhancement Feature:

يمكن لمصمم قاعدة البيانات حماية البيانات، أو لنقل زيادة وتعزيز سلامة قاعدة البيانات، وبالتالي سلامة البيانات وتكاملها، بمجموعة كبيرة من ما يمكن أن نسميها بالاحتياطات، أو الشروط، هذه الشروط كلما كانت مطبقة على قاعدة البيانات، كانت سلامة البيانات أكثر ضمانا، ومن هذه الشروط:

١) فرض وجود البيانات required data
٢) قيود أو شروط المجال domain constraints
٣) سلامة الكيان (الكينونة) entity integrity
٤) السلامة المرجعية referential integrity
٥) الشروط العامة general constraints

١) فرض وجود البيانات required data:
أحيانا تكون هناك ضرورة ماسة لتجنب القيمة NULL، في أحد صفوف الجدول، ولأجل هذا يتم فرض وجود البيانات عن طريق التعبير NOT NULL، وهذا التعبير يستخدم في جملة CREATE TABLE التي يتم بواسطتها إنشاء الجدول، وتحدد لأحد الحقول التي لا نرغب أن تكون أي من سجلاته/صفوفه فارغة، وهكذا نظمن عدم قبول أي تعديل أو إضافة إلا إذا كانت تلك القيمة غير فارغة.
مثال ذلك في المفاتيح الرئيسية P.K، حيث أن القيمة الخالية تعني أن السجل غير موجود، وهناك أمثلة أخرى تتعلق ببعض البيانات، مثل الراتب SAL في جدول الموظفين، فلا يمكن أن يعمل موظف دون أن يحدد له راتب.
وهذا الشرط متعلق كما هو واضح بعمليات التعديل والإضافة فقط، إذ لا يمكن لعملية الحذف أن تنتهك سلامة هذا الشرط.

٢) قيود أو شروط المجال domain constraints:
نعلم أن المجال DOMAIN هو مجموعة القيم التي يجب أن لا تتجاوزها قيم حقل معين، على سبيل المثال، درجة مقرر لطالب في قاعدة بيانات الطلاب، لا يمكن أن تقل عن صفر ولا أن تزيد عن مائة، فنطاق أو مجال هذا الحقل هو (٠-١٠٠)، ويتم التأكد من ذلك في لغة SQL، عن طريق الدالة CHECK()، التي نضع داخلها اسم الحقل والنطاق الذي يجب عدم تجاوزه.
وكالشرط السابق فإن عمليتي إدخال صف جديد وتعديل صف موجود هما اللتان تنتهكان هذا الشرط.

٣) سلامة الكيان (الكينونة) entity integrity:
يتم المحافظة على سلامة مثيل أو وحدة الكيان (Entity Instance)، عن طريق تمييزه عن غيره من الوحدات، وذلك بواسطة تخصيص مفتاح أساسي يميز كل وحده عن الأخرى ضمن نفس الكيان، وهذا هو المفتاح الرئيسي الذي يجب أن يتميز بخاصيتين أساسيتين هما:
١/ أن لا يكون فارغا Not null، ٢/ أن يكون فريدا Unique أو وحيدا فلا يسمح بتكراره.
وهذا الشرط أيضا يتم انتهاكه عادة بعمليتي إضافة سجل جديد، أو تعديل سجل موجود، ولا يتأثر بعملية الحذف.

٤) السلامة المرجعية referential integrity:
عندما يكون هناك مفتاح أجنبي FK **في جدول ما** (يسمى بالجدول الإبن)، فإن من السليم أن يشير دائما **إلى مفتاح أساسي** PK **موجود في الجدول المرجعي (أو الجدول الأب)**، ولكن لو حدث تعديل أو حذف لسجل يحتوي على مفتاح أساسي، في الجدول الأب، مع بقاء المفتاح الأجنبي الذي يشير إليه في الجدول الإبن، **فإن المشكلة الناتجة عن وجود مفتاح أجنبي لا يشير إلى مفتاح أساسي موجود، تسمى بمشكلة السلامة المرجعية**، وتحدث أيضا عند إضافة مفتاح أجنبي إلى صف في جدول إبن، دون أن يكون هناك مفتاح أساسي يقابل ذلك الصف في الجدول الأب.
ويتم حل هذه الإشكالية في لغة SQL عن طريق مجموعة من الإيعازات، التي تستخدم إما عند إنشاء الجدول، أو بإجراء تعديل على بنية الجدول باستخدام ALTER TABLE، ووضع أربعة إيعازات، تمثل الاحتمالات المطبقة عند إجراء أحد عمليات معالجة البيانات DML، وخاصة التعديل والحذف، وتحديد الإجراء المناسب عند محاولة انتهاك هذا الشرط، وهي:

١/. تكرار التعديل أو الحذف وذلك بالإيعاز CASCADE.
٢/. إعادة قيمة FK المنتهك إلى الفراغ NULL، بالإيعاز SET NULL، ومشكلة هذا الخيار تظهر في حالة كان جزء من المفتاح الأساسي للجدول الإبن، أو في حالة لم يكن كذلك، ولكنه كان مشروطا بشرط ضرورة الوجود، أي NOT NULL.
٣/. تحديد قيمة تلقائية عند تعريف حقل الـ FK **في الجدول الإبن**، ويستخدم الإيعاز SET DEFAULT لطلب هذا الخيار، أي عد بقيمة المفتاح الأجنبي المنتهك إلى قيمته التلقائية.
٤/. والخيار الرابع هو منع عملية الانتهاك نفسها، وذلك عن طريق الإيعاز NO ACTION، والذي يقصد به توقيف عملية التعديل أو الحذف على الجدول الأب، لحماية المفتاحين الأجنبي والأساسي، في كلا الجدولين.

٥) الشروط العامة general constraints:
وهذا الشرط يستخدم مجموعة كبيرة من القيود، هي الشروط الفائضة عما سبق، أو غير المتعلقة بالشروط السابقة، ومنها شرطي المجال والقيمة الفريدة المستخدم لقيم مختلفة عن الحقول المفتاحية، وذلك بالإيعازين (CHECK و

UNIQUE)، وهناك كائن يسمى ASSERTION يعرف في لغة SQL بطريقة تجعله يراقب مجموعة كبيرة من الإنتهاكات، ويضع الحلول المتاحة، يمكن للطالب أن يقدم تقريرا حوله، ويقوم بتطبيقه في المعمل.

ويلخص ما ذكرناه بالجدول التالي:

م	**الشرط/القيد Constraints**	**عملية DML المؤثرة**	**أسلوب تعزيز سلامة البيانات في SQL**
١	required data	INSERT,UPDATE	NOT NULL
٢	domain constraints	INSERT,UPDATE	CHECK()
٣	entity integrity	INSERT,UPDATE	NOT NULL + UNIQUE
٤	referential integrity	ALL DML	CASCADE/SET NULL or SET DEFAULT
٥	general constraints	INSERT,UPDATE	CHECK/UNIQUE or ASSERTION

المشهد أو المنظور Views:

هو نتائج آلية لاستعلام أو أكثر يتم تطبيقها على جدول أو أكثر لإنتاج جدول مؤقت جديد يسمى بالمنظور View.

أي المنظور هو جملة استعلام محفوظة داخل جدول مؤقت، وتكون مرتبطة عادة **بجدول أساسي أو أكثر من جدول، حسب جملة الاستعلام المستخدمة**، (المقصود بالاستعلام البيانات الناتجة عن استخدام الجملة select، كما سيأتي).

ويعرف المنظور أيضا بأنه: جدول افتراضي غير موجود بالفعل في قاعدة البيانات، ولكنه يولد أو ينتج عند الطلب، وقت الاحتياج.

في لغة تعريف البيانات DDL يتم إنشاء المنظور باستخدام الجملة Create View والتي لها الشكل العام التالي:

```
CREATE VIEW ViewName [ (newColumnName [,...]) ]
AS subselect ;
```

حيث أن **subselect** يقصد بها جملة الاستعلام المستخدمة لإنشاء المنظور، ومن الواضح ضرورة أن تكون **أسماء** الحقول في تعريف المنظور متطابقة مع أسماء الحقول في جملة الاستعلام الفرعية subselect، من جهة العدد ونوع البيانات، كما أنه يمكن عدم تسمية أي حقول للمنظور، حيث ستأخذ حقول المنظور نفس أسماء حقول الاستعلام.

مثل: يتم هنا توليد منظور بالاسم **Manager3Staff حسب جملة الاستعلام الموضحة والتي تستخدم الجدول** staff :

```
CREATE VIEW Manager3Staff
    AS  SELECT *
        FROM Staff
        WHERE branchNo = 'B003';
```

مثل: **حذف المظور، وحذف الـ** view يتم بنفس طريقة حذف الجدول:

```
    DROP VIEW Manager3Staff;
```

هذه التقنية يستخدمها مدراء قاعدة البيانات DBA، بطرق مختلفة، من أجل بعض الميزات، ولكن للمنظور أيضا بعض العيوب.

مميزات استخدام المنظور Advantages of Views

١) استقلالية البيانات Data independence:

حيث يستطيع مدير قاعدة البيانات فصل بعض البيانات وأجراء بعض المعالجات عليها، والتعامل مع المنظور **وكأنه** جدول مستقل.

٢) **التزامن** Currency:

يمكن الاستفادة من المنظور في رفع مستوى تزامن البيانات، في قواعد البيانات الموزعة distributed database.

٣) رفع أمنية البيانات Improved security:
إن اختلاف صلاحيات المستخدمين يعني بالضرورة اختلاف الكائنات أو طرق التعامل مع كائنات قاعدة البيانات، واستخدام كائن المنظور View يساعد في ذلك، مما يساعد في رفع أمنية البيانات.
٤) تقليل تعقيدات التعامل مع بعض الاستعلامات Reduced complexity.
٥) المحافظة على سلامة البيانات Data integrity والرفع من ميزات السلامة حسب الشروط الموضحة أعلاه.

عيوب استخدام المنظور Disadvantages of Views

١) محاذير التحديث Update restriction:
وذلك بسبب صعوبة تحديث البيانات عبر المنظور، حيث أن هناك شروط كثيرة على الاستعلام المستخدم لإنشاء المنظور، حسب نوع المنظور وجملة الاستعلام المستخدمة، ويمكن للطالب البحث في هذا الموضوع كتمرين.
تمرين: قم بالبحث في موضوع Update restriction موضحا متى يمكن التحديث في المنظور ومتى يصعب ذلك؟
٢) محاذير بنية المنظور Structure restriction: كذلك الحال بالنسبة لإنشاء المنظور ومواصفاته.
٣) الأداء Performance:
إن الاحتفاظ بالمنظور في قاعدة البيانات، يعني زيادة المساحة التخزينية لها، مما يؤثر على أدائها، أما عدم تخزين المنظور، فيعني ضرورة إنشاءه كل مرة نحتاج إليه، وهذا يؤثر أيضا على الأداء.
إن وجود العيوب سابقة الذكر، لا ينفي فائدة المنظور في إدارة قاعدة البيانات، ولكن يجب الأخذ بعين الاعتبار تلك الميزات والعيوب.